Bellcomm

955 L'Enfant Plaza North, S.W.
Washington, D. C. 20024

B72 03019

date: March 31, 1972

to: Distribution

from: D. A. Corey, D. N. Sakolosky, E. W. Radany

subject: A Computer Program to Aid Skylab Earth Resources Experiment Planning - Case 610

ABSTRACT

The multitude of data which must be handled by the mission planner in his planning studies of the Skylab Earth Resources Experiment Program makes such studies most formidable. This memorandum documents a computer program designed to handle much of the necessary data for the planner thus making such studies more feasible. The program is designed to be used by the planner in an interactive time sharing environment. The documentation describes the data base, the theory and workings of the program, its use, its known errors, and the necessary details of running it.

(NASA-Cr-126940) A COMPUTER

TABLE OF CONTENTS

Bellcomm

955 L'Enfant Plaza North, S.W.
Washington, D. C. 20024

date: March 31, 1972

to: Distribution

from: D. A. Corey, D. N. Sakolosky, E. W. Radany B72 03019

subject: A Computer Program to Aid Skylab Earth Resources Experiment Planning - Case 610

MEMORANDUM FOR FILE

INTRODUCTION

The operating aspects of the Skylab Earth Resources Experiment Program (EREP) differ significantly from the other scientific investigations to be carried out on Skylab in that a basic facility is being supplied for use by a large number of investigators rather than just a few. Each of the investigators is interested in different data taken under widely-differing conditions. For example, a study of a candidate set of 161 investigations, or proposals, (the same set used in an MSC mission sizing study) revealed some 891 schedulable objectives. Many of these objectives require multiple passes over the ground site to be satisfied. The proposals referenced 556 different, though in many cases, overlapping, ground sites. The data required is to be taken under a wide variety of lighting constraints, cloud cover constraints, and in various months throughout the year. Each objective requires that a certain set of sensors be used in the various modes in which the sensors can operate. In addition, many of the proposals have special constraints. Some require data obtained only on an ascending (heading in a northeasterly direction) pass across the ground site. One requires a night pass occurring within one day of a daylight pass. The mission planner must combine this complex set of requirements with the trajectory data, and with the other constraints placed on the crew work schedule. The uncertainty of local weather in the multitude of areas of interest further adds to his problem. We concluded that the planning task was too formidable to be efficiently done by hand. Certainly, the numerous studies one would like to perform would be severely limited. Consequently, it was decided to build a computer program which would take care of most of the bookkeeping and data handling required, and hopefully free the planner of most of the tedious labor involved.

This memorandum describes that program as it exists now. While there are desirable additional features and capabilities not currently implemented, it is felt that the program is advanced enough to be useful in mission planning studies and, perhaps, in the real time flight control environment. The demise of Bellcomm precludes further development and use by Bellcomm, but it is hoped that NASA will find it useful enough to justify the usual pangs of picking up somebody else's program. Since the current situation was known at the program's inception, every effort was made to keep its structure and coding as straightforward as possible.

The program was designed to be used by the planner, as opposed to a programmer, in an interactive time-sharing environment. The user is only required to remember a few commands and some flexibility in their proper spelling is allowed. He is requested, more or less in English, to supply necessary problem data at the time it is required. Where a particular format for user supplied data is expected, he is reminded of the correct format. Some data is less frequently supplied by the user so provision is made for him to be reminded of proper spelling if he so desires. The net result, it is hoped, is that the program is relatively simple to operate and the user's thought processes may be given over to more important things. None of this, of course, obviates the need for the user to have a basic understanding of what the program is doing when he gives it a command. He must have the understanding to properly operate the program and to correctly interpret the results.

Operation of the program in a batch mode is somewhat awkward in that the data deck must be carefully stacked and program requests for data or responses must be anticipated. There are certain situations, however, where batch operation is worth the trouble - where voluminous displays of data are required for example.

With the exception of one subroutine, the entire program is written in Fortran and uses standard UNIVAC library routines. It is currently structured to be run on a UNIVAC 1108 under the Exec 8 operating system. It currently requires approximately 13000 words for program storage and about 45000 words for data storage. Further segmentation and linked operation of the program is, of course, possible if the storage requirements are too large for a particular environment.

The non-Fortran subroutine, CINOTE, supplies a convenience feature and is not actually required by the program. It allows interruption of a lengthy data display, in the event the user gets tired of looking at it, and returns control to the user. It is written in 1108 assembly language and a listing is included with the other listings.

DATA BASE

A group of files which catalog the requirements of the various proposals, describe the ground sites, and specify the parameters of each revolution of the trajectory comprise the main data base. The contents and organization of the files in each of these categories are described below.

Proposal Requirements

The data file titled EREPD describes, in a coded format, the requirements of the majority of the proposals for performing earth resources investigations presently under consideration. EREPD is the raw data file constructed directly from a reading of the proposals. Several proposals were omitted for the following reasons: (a) the proposer merely required data from some other investigation, (b) the associated ground site would only be specified shortly before the mission, or would be specified in real time.

The file EREPD is constructed to be read by a namelist read directly into an array, EREP, dimensioned 15, n; where n is the number of proposal subobjective (FO's) to be considered. Accordingly, the first card image in EREPD is "Δ$EREPD", the second card image is ΔEREP =, and the last card image is "Δ$END". (The Δ indicates a blank in column 1.) Each card image in between contains a blank in column 1 followed by fifteen floating point numbers. A comma follows each number.

The sequence and format of each of the entries in a row (or card image) are described below.

Entry 1 - proposal number, assigned by NASA. Most usually a three digit number; e.g. 052., 356., etc. Subsectioned proposals are encoded with a four digit number with the proposal sub-section number appearing as the thousands digit; e.g. proposal 540-3 is indicated by "3540."

Entry 2 - flight objective number. A flight objective is defined to be an individual, schedulable event necessary for the performance of a particular proposal. For example, a particular proposal specifies that two passes over a prescribed site are required; one a daylight pass in May, and the other a night pass in November. One daylight pass over the site occurring specifically in May would be denoted as flight objective #1 of this proposal, and a night pass occurring specifically in November would be flight objective #2. Multiple sites, variable instrument combinations, different seasonal requirements, etc., associated with a proposal result in the designation of multiple flight objectives.

Entry 3 - site number. This number denotes the ground site associated with the particular proposal. The geographic description of each of the sites referenced in the proposals is contained in a file titled SITE which is described later.

Entry 4 - discipline. A one-digit number which encodes the discipline into which the proposal is categorized; viz. 1 = agriculture/forestry, 2 = ecology, 3 = geography, 4 = geology, 5 = hydrology, 6 = interpretation techniques development, 7 = meteorology, 8 = oceanography, 9 = sensor technology.

Entry 5 - sensors required. A five digit number, each digit of which is either "0" or "1", used to encode the sensor requirements for the particular flight objective. The units digit indicates S-190, the tens digit indicates S-191, etc. The digit "1" indicates that the instrument is required, and the digit "0" indicates that the instrument is not required.

Entry 6 - seasonal requirements. A four digit number which indicates the extent of the calendar months within which a flight objective may be satisfied. The tens and units digits indicate the latest acceptable month for satisfying the flight objective; the thousand and hundreds digits indicate the earliest acceptable month. A flight objective for which a pass at anytime between May and September was acceptable would thus have the season entry "0509."; a flight objective for which a pass between November and March was acceptable would have the season entry "1103."; a flight objective which required a September pass would have the season entry "0909.".

Entry 7 - pass type. This four digit number describes the pass, or passes, required to satisfy a particular flight objective. The units digit may be either "0", "1", or "2" with the following interpretations:

"0" - arbitrary, unrelated pass(es) over a site is required.

"1" - repeating passes at "ab" day intervals are required, where "ab" are the hundreds and tens digits of the pass type descriptor.

"2" - complete coverage of the site is required.

The thousands digit indicates either the number of passes required when the units digit is "0" or "1", or the number of complete coverages required when the units digit is "2". A requirement for two arbitrary passes is indicated as "2000.", "2101." indicates that two passes separated by a ten-day interval are required, and "2002." indicates that two complete coverages of the site are required (the number of passes to achieve two complete coverages is unspecified). In some cases multiple repeat coverages are required. Two total coverages 20 days apart for example, would be coded 2202.

Entry 8 - special note. This is a number which refers to a separate list of unique requirements which could not be reasonably formatted otherwise. Examples are: Note #3 (Proposal 380), "One night pass ideally to occur within one day of a daylight pass; Note #11 (Proposal 446), "Lighting to give minimal sun glint". When a note is not required, "0" was entered.

Entries 9 and 10 - lighting data. Entries 9 and 10, respectively, give the minimum and maximum acceptable sun elevation angles at the site in degrees.

Entries 11 and 12 - cloud cover data. Entries 11 and 12, respectively, give the minimum and maximum allowable cloud cover over the site in percent of complete overcast; no reference is made to cloud type.

Entries 13 and 14 - snow cover data. Entries 13 and 14, respectively, give the minimum and maximum acceptable snow cover in percent of the site area which is snow covered. Required snow depths, if any, are not recorded.

Entry 15 - flight objective weight. This entry gives the weight, within each individual proposal, assigned to the corresponding flight objective. The total of the weights of all the flight objectives of each proposal is 100; i.e. all the proposals are considered to be of equal importance, though the individual FO's may be unequally weighted if the proposer so indicated.

The program currently takes no action as a result of a special note (or entry in item 8). The following is a list of the meaning of the notes.

#1, Prop. 057, "Sites provided during mission in real time".

#2, Prop. 357, "Pass 1 at 1000, passes 2-9 between 0600 and 1200, pass 10 at 1500". (All times are local.)

#3, Prop. 380, "One night pass ideally to occur within one day of a daylight pass".

#4, Prop. 423, "Descending passes preferred".

#5, Prop. 541, "Ascending pass required".

#6, Prop. 541, "Descending pass required".

#7, Prop. 583, F01, "Look for glory or glitter and non-glory or non-glitter; corn".

#8, Prop. 583, F02, "Ground track crossover points at 1.5, 3.0, or 4.5 hour intervals".

#9, Prop. 583, F03, "Look for glory and non-glory; wheat".

#10, Prop. 070, "Require observation of spring tides".

#11, Prop. 446, "Lighting to give minimal sun glint".

#12, Prop. 571, "Frequency: Pairs either at 2-3 day intervals or 3 passes during SL-2, 3 passes during SL-3, and 4 passes during SL-4." (The EREPD array entries assumed the second option.)

#13, Prop. 587, "Imagery of earth limb".

#14, Prop. 433, "Preferred: descending passes over areas 1, 2, and 3. First alternate: ascending passes over areas 2, 3, and 4. Second alternate: descending passes over areas 4 and 5".

#15, Prop. 464, "Specific geographic area supplied in real time".

#16, Prop. 052, "Pass between 1200 and 1600". (Corresponding sun angles depend on latitude and season.)

#17, Prop. 052, "Pass between 1100 and 1300". (Corresponding sun angles depend on latitude and season.)

#18, Prop. 441, "One pass, pre-dawn". (Sun elevation angle information precludes distinguishing between pre-dawn and post sun-down.)

#19, Prop, 441, "One pass, after sun-down".

#20, Prop. 457-1, "Two passes with S-193".

#21, Prop. 540-6, "Lock on and track site 541 with S-191 during all overflights".

The proposal requirements data are input to the main EREP program from the data file ERDATA. ERDATA is generated from EREPD by the auxiliary routine MSORT. The EREPD file is read via a namelist read, and each FO is sorted into a mission category as follows:

Category	Description
2	FO's which can be accomplished only during the SL-2 mission
23	FO's which can be accomplished only during the SL-2 mission or the SL-3 mission
24	FO's which can be accomplished only during the SL-2 mission or the SL-4 mission
234	FO's which can be accomplished during any of the three missions
3	FO's which can be accomplished only during the SL-3 mission
34	FO's which can be accomplished only during SL-3 or SL-4
4	FO's which can be accomplished only during SL-4

Any FO not falling into one of the categories is ignored. The sorted FO's are then placed into a proposal information array by mission category in the order 2, 23, 24, 234, 3, 34, 4.

The following arrays are output in binary to file ERDATA: (1) a 7x2 integer array where (k,1) and (K,2) indicate the start and end of the data for the Kth mission category in the proposal array and (2) the 15x1000 floating point proposal array where

(J,K), J=1,15 is the proposal information for the Kth entry in the proposal array.

Ground Site Description

Throughout the proposals, ground sites were generally specified in one of three fashions. First, many proposals explicitly indicated the geographic coordinates defining the site; such data were recorded verbatim. Secondly, certain proposals referenced designated Earth Resources Experiment Package Test Sites; the coordinate data for these sites were obtained from the February 1971 issue of the EREP Users Handbook. Lastly, many proposals merely described the required sites verbally; viz. the Sahara, Mediterranean shore of Israel, names of cities, names of Calderas, etc. Coordinate data for such sites were variously obtained from the following sources:

(1) Rand McNally Road Atlas, 47th Annual Edition

(2) The Times Atlas of the World, Mid-Century Edition, Houghton Mifflin Company, Boston 1957.

(3) The National Atlas of the United States of America, United States Department of the Interior, Geological Survey, Washington, D.C., 1970.

(4) The Encyclopedia of Oceanography, Rhodes W. Fairbridge, Reinhold Publishing Co., New York, 1966.

(5) Volcanic Landforms and Surface Features, J. Green and N. M. Short, Springer-Verlag, New York, 1971.

The raw ground site data are maintained in a file element named NEWDAT. This file lists each site number and the latitude and longitude (both in degrees) of the corner points of each site.

The data in NEWDAT is organized to be read by a namelist read directly into two arrays in the auxiliary program SITES. The first card image in NEWDAT is Δ$INEREP and the last card image is Δ$END where the "Λ" indicates the first column blank. All card images in between must contain a blank in column 1. The site data, which is to be read into the array EREP, is preceded by the card image ΔEREP=. Subsequent card images contain the site corner point values - expressed as the latitude (in degrees) followed by the longitude (in degrees) of the point. The set of points for one site is separated from the set of points for the next by the two numbers: 9999.0, 9999.0,. (A comma must follow every data value in namelist formatted input.) The end of the last set of points is indicated by the two numbers: -9999.0, -9999.0,. The remainder of the data in NEWDAT is a list of the site numbers assembled in the same order as the sets of corner points. This data will be read into the array EREPNO and so is preceded by the card image ΔEREPNO=.

The auxiliary program SITES is located on program file (and element) EREP.SITES. SITES reads the site data from NEWDAT, converts it to a format readable by the main program (subroutine COMMON reads it) and outputs it onto the data file LATLON.

LATLON contains a counter and two arrays written out in binary. The counter is the integer value of the highest numerical value assigned as a site number. One array is an integer array, dimensioned 2x600. Entry (1,K) indicates the start location in the array of corner points at which points for the Kth site are defined. Entry (2,K) indicates the number of points which define the Kth site. The other array contains the site corner points, is floating point, and is dimensioned 2x5000. Entry (1,L) is the latitude of a point, and entry (2,L) is the longitude of a point (degrees). Note that if either the total number of sites or the highest numerical value of a site exceeds 600, or if more than 5000 corner points are required, the dimensions of the arrays should be enlarged - in subroutine COMMON as well as in SITES.

Trajectory Description

The trajectory data for the current nominal launch conditions, generated by a modified version of the Bellcomm Apollo Simulation Program and including the three missions, are stored in binary on file NEWANO. Routine MISION on program file EREP reads the data from NEWANO, gleans all data necessary for input to the EREP program, and stores it in binary on data files SL2DAT, SL3DAT, and SL4DAT. These data files contain trajectory information for SL-2, SL-3, and SL-4 respectively. The format of NEWANO will not be described because the file is basically unnecessary. It was available to Bellcomm and was therefore referenced in lieu of generating another set of trajectory data in the formats described below for SL2DAT, SL3DAT and SL4DAT. It is assumed that any user generating his own trajectory information can create the necessary data files in the appropriate format initially.

The start and end times for trajectory data for each mission are specified in the body of the MISION routine. Each data file created by MISION contains the following written out as one binary record:

(1) An integer specifying the number of revolution descriptions included on the file

(2) An integer indicating the number of the first rev for which the absolute magnitude of beta exceeds the maximum allowed value (currently 50°). (Beta is the angle between the solar vector and the orbit plane.)

(3) An integer indicating the number of the last rev for which the absolute magnitude of beta exceeds the maximum allowed value. (If the absolute value of beta never exceeds the maximum, the values for the integers in (2) and (3) are set to zero).

(4) A floating point array dimensioned 5x700 where, for the Kth entry,

(1,K) is the rev number

(2,K) is the longitude (degrees) of the ascending equatorial crossing.

(3,K) is the value of beta at the ascending equatorial crossing

(4,K) is the elapsed time (in days from midnight, Dec 31) of the ascending equatorial crossing

(5,K) is the value of theta at orbit noon

No data is included for revs for which the absolute magnitude of beta is greater than the maximum allowed.

Revolutions are counted from 50°S latitude to 50°S latitude. In terms of theta/omega coordinates, a revolution from 50°S latitude to 50°S latitude corresponds to a theta range from -90° to 270°. This definition was selected for convenience so that passes over all sites presently recorded in the LATLON file can occur on one revolution. That is, if revolutions were counted from an ascending equatorial crossing to the next ascending equatorial crossing, for example, overflights of sites which straddle the equator would have to be counted as occurring partially in revolution "N", and partially in revolution "N+1". This bookkeeping complication has thus been avoided.

PROGRAM OPTIONS, OPERATION, AND USE

The user directs the operation of the program by specifying various options to be executed. Before discussing the various options and the operation of the program, a description of the few rules governing the user's communications with the program will be helpful.

Yes and No responses: In several cases, the program requires a yes or no type response from the user. For example, the program may display the settings of several program variables and then ask the user if they are set to the values he wants by typing "OK?". The user signifies a negative response by typing "no" and a carriage return. Any other response, including a simple carriage return, will be interpreted as "yes".

NAMELIST INPUT: There are several situations where the user may wish to change the values of one or more program variables and the system namelist routine is used to read the new values. Where the program desires input in this form, it will so indicate by saying for example: "SUPPLY NEW VALUES NAMELIST NAME": where NAME is the name of the list to be used. The user then types, starting in column 2, $NAME (or whatever the namelist name is), followed by at least one blank, the name of the variable to be changed, an equal (=), the new value desired, a comma and finally "$END". As many variable names, =, and new values as are desired may be inserted between the $NAME and the $END card but a comma must follow each set. Multiple lines may be used and no more than 72 columns should be used on any one line. If the same variable is set more than once, the last value will prevail. If the user changes his mind and wants no values changed a simple $NAME $END will work. The following is a typical example used at program initiation time or under the CHANGE option.

```
 $CONTRL APASDE=120., MCATDE=2, (carriage return)
 MISSION=3, $END (carriage return)
```

= INPUT: The program also requires some direct input from the user which may be either data or a Hollerith word. (The context of the question should help the user distinguish between these and the questions requiring a yes or no input.) In these cases, the required information is simply typed in directly. Only the questions "OPTION=" and "PRINT OPTION=" require Hollerith input. In the cases where integer data is required, the program will indicate the proper format. If, for example, the question "REV NO.? (I4 FORMAT)" is asked, the program expects a four digit number. If the user wishes to enter the value 61, the leading zeros must be typed - 0061. Blanks may be substituted for leading zeros.

When the program is ready to accept an option request, the phrase "OPTION=" will be typed by the computer. The names, alternate spellings and descriptions of the available options follow.

EVALUATE OR EVAL OR E

The evaluate option will cause the program to evaluate a designated revolution in order to determine which FO's can be satisfied on that revolution. The FO's which can be at least partially satisfied on the rev will be listed in the SCORE array together with the theta limits of the portion of the rev satisfying the FO and the constrained and unconstrained weight accomplished on the revolution. The decision as to whether the FO should be listed is based on whether the unconstrained weight accomplished is non zero, not the constrained weight.

Following entry of the EVALUATE option, the user will be requested to supply the revolution number to be evaluated by the statement "REV NO.? (I4 FORMAT)". The current trajectory data numbers the revolutions with revolution 1 commencing the first time theta = 270 degrees (=-90°) following SL-1 launch - that is, at the first southern-most extent of the SL-1 orbit following launch. Revolutions are counted as nodal revolutions from $\Theta=-90°$ to $\Theta=+270°$. The I4 format statement indicates that the program expects a four digit number - leading zeros should be supplied as zeros or blanks. Normally, no further output is to be expected from the evaluate option.

MAXWGT or M

The MAXWGT option searches through the most recently evaluated revolution to find the optimum location for the Z-local vertical pass in terms of maximizing the constrained weight achieved by the pass. The user is allowed to specify the length of the pass and the sense in which it is optimized. He may also constrain the search to a specified segment of the revolution if he wishes.

After the user types "maxwgt", the program will display the default values of ARCPAS, MCATEG, THMIN, and THMAX and ask if these are "OK". ARCPAS is the length, in degrees, of the Z local vertical pass. The value of MCATEG will control the sense in which the optimization is performed. MCATEG can be equal to 2,3,4,23,24,34, or 234. The program in its optimization, will ignore those FO's which have mission categories which include a number not found in MCATEG. For example, if MCATEG=23, FO's in the following categories will not be included in the search for the optimum: 4,24,34, and 234. All FO's will be included if MCATEG=234. THMIN and THMAX are the miminum and maximum values of theta which will be considered for starting the Z-local vertical pass. If a "no" response is received, the program types "SUPPLY VALUES, NAMELIST MAX", indicating that new values should be entered using namelist $MAX.

The program next displays some summary results of the optimization - the rev number, the optimum value of theta at the start of the pass (THSTRT), theta at the end of the pass (THEND), and the constrained and unconstrained weights achieved by the pass (WGTMXC and WGTMXU). These weights do not include FO's which would be accomplished on the pass but which do not fall in a category included in the optimization. (A complete display of the achievements may be obtained by calling for print option REVSUM.) Note that the MAXWGT option results in values actually being written into THSTRT and THEND. These pass theta limits will be retained through subsequent options unless expressly overwritten by the user. A particular set of theta limits may result in the extent of the useful pass being shortened but not eliminated for one or more FO's. In these cases, either at least ARCMIM degrees of the FO pass must remain or at least one-half the arc length of the possible pass must remain in order for the FO pass to be counted (see subroutine PASBAD).

The answer provided by MAXWGT is not necessarily unique. There may be many or a wide range of places to start the Z local vertical pass which will achieve the same weight. The user can explore for these by using the print suboption REVSUM.

SCAN or S

The SCAN option automatically executes the EVALUATE and the MAXWGT options for every revolution over a prescribed set of days and determines the n best in terms of the maximum constrained weight accomplished.

On receiving the SCAN option, the program asks the user to supply the day number, (in terms of January 1, at Greenwich = day 0) of the first day of the scan by printing "FIRST DAY OF SCAN =? (I4 FORMAT)". The user supplies the properly formatted answer and then responds similarly to a request for the last day of the scan. These day numbers may be identical if only one day is to be scanned. The program next asks "OMEGA LIMITS DESIRED?" If the user knows he wants to consider only certain parts of the world, e.g., U.S. passes, he can substantially speed up the program by limiting the allowable range of omega of the revolutions considered. An affirmative response will evoke the request "TYPE IN WESTERN OMEGA LIMIT." which the user then supplies as a floating point number in degrees. This is followed by "TYPE IN EASTERN OMEGA LIMIT" which the user similarly answers. The limits may be entered either with the convention 0 through ± 180° or 0 through 360°. In either case, the angle is measured positive eastward. (Note that it is permissible for the Eastern limit to be numerically smaller than the Western limit.) If the user is particularly interested in foreign areas, applying omega limits will generally be required since the heavy weighting given U.S. passes may swamp out the weight achieved on foreign passes.

Next the program asks the user to "INPUT NUMBER OF ORBITS TO BE RETURNED (I2 FORMAT)". Recall that the scan option will display the "n" best revs. This question allows the user to specify "n".

Finally the program displays the same data displayed following the MAXWGT option and asks if it is OK?. The user responds just as he would for the MAXWGT option.

The computer then scans the revolutions, determines the best ones and displays a summary of the scan by printing the rev number, THSTRT and THEND for the best Z local vertical pass location, and the constrained and unconstrained weight achieved by the pass.

The user should keep in mind that when the SCAN option is completed, the information in the SCORE matrix and in THSTRT and THEND pertains to the last revolution evaluated which may or may not be one of the listed best. In order to further study one of the best revolutions, the evaluate option should be called for that revolution.

ACCEPT or A

The ACCEPT option simulates the effect of actually performing a Z local vertical pass by adding the accomplishments of the pass to what has previously been accomplished (accepted). For details, see the descriptions of the OMECOV, PASS, TOTSCR, and PASDAT arrays in the section entitled WORKING DATA ARRAYS AND VARIABLES.) After receiving the ACCEPT option, the program asks: "ACCEPT REV = IREV, PERWT = "PERWT" OK?", where IREV is the number of the revolution most recently evaluated and "PERWT" is the current value of PERWT. A negative response to this question cancels the ACCEPT option. (IREV can be changed via the EVALUATE option and PERWT can be changed via the CHANGE option.) If a positive response is received, the program displays the values of THSTRT and THEND and asks "OK?". A negative response will enable the user to change the theta limits of the pass prior to acceptance. Otherwise the current values are used to update the required arrays and variables.

CHANGE or C

The CHANGE option affords the user the opportunity to change a number of control variables via a namelist input. The user is first given the opportunity to see the current values displayed before he is requested to supply the new values. The variables which can be changed are APASDE, ARCMIN, IDEVIC, IERSAV, IEROLD, MCATDE, MISION, PERWT, THMADE, and THMIDE. The definitions of these variables will be found in the section labeled VARIABLES IN COMMON SECTION [CONTRL].

In essence, the CHANGE option, with the display, is invoked automatically at program initiation. At any other time, changing IEROLD via the CHANGE option will be ignored. IEROLD is the number of the data unit on which the results of a previous session are stored. Since the constrained weight accomplished by a revolution is a function of what has already been accepted, attempting to add accomplishments from a prior session to the accomplishments of the current session would simply produce garbage and is, therefore, not allowed. All other parameters may be meaningfully changed either at the beginning or during the session and these changes are heeded.

PRINT or P

The PRINT option enables the user to have printed any of a number of data displays. He selects which particular display by entering the name of a print option. The names and a brief description of the currently available print options are given below. In a few cases, the print options require additional data from the user. The program will request this data if required.

In a few cases, the amount of output printed depends on the parameter IDEVIC. If IDEVIC = PRNTR, substantially more output will be provided than if IDEVIC=TERMNL.

In some cases, the user may abort the printing (by depressing the ATTN key on an IBM 2741 Terminal) (if the CINOTE subroutine works at the particular installation). The program will skip the detailed print, continue processing quietly, print a short summary, and then return control to the user. If CINOTE is not working, depressing the ATTN key may abort the program (at least on Bellcomm's UNIVAC 1108 it does). There seems to be a problem with CINOTE when using the teletype terminal at Bellcomm. Depressing the BREAK key, which should be equivalent, simply suspends processing. A carriage return then causes it to resume as though nothing happened. The PRINT OPTIONS:

ALDONE: Prints a list of the FO's which have been completely accomplished by the revolutions accepted so far (Abortable)

ERESUM: For every proposal which can be at least partially worked on the currently considered mission (the list depends on the value of the parameter MISION) the constrained and unconstrained weight accomplished on the most recently evaluated revolution, and the accumulated accomplished weight and the total possible weight are printed. (See Samples of Printout.) If IDEVIC=TERMNL, only the total weights for each proposal are listed. IF IDEVIC = PRNTR, every FO is listed. A summary is printed at the end. (Abortable)

MISSUM: Prints a summary of the accomplishments accepted so far for the mission currently being considered. The weight accomplished (WGT SAT), the total weight possible (TOT WGT POS), the number of FO's which have been at least partially satisfied (NON ZERO FOS), and the total number of FO's in the mission category (TOT FOS) are printed by mission category. Totals are printed as well.

NOWORK: Provides a list of every FO on which nothing has yet been accomplished. A summary is printed at the end. (Abortable)

PRGSUM: Essentially identical to MISSUM except data for the mission categories not included in the currently considered mission are printed as well.

PASSUM: Provides a summary of the passes which have been accepted so far. For each pass accepted, the following is printed: The revolution number (REV NO), the geographic longitude of the ascending equatorial crossing of the revolution (OMEGA), the GMT of the start of the Z local vertical pass expressed as month (MO), date (DAY) and hour (HR), theta at the start (THSTRT) and end (THEND) of the pass, and the sun elevation angle at the start and end of the pass.

PRPSUM: For every FO of a specified proposal, the constrained and unconstrained weight which can be accomplished on the most recently evaluated revolution, the total weight accomplished (accepted) so far and the total possible weight are printed. Required additional input - the proposal number.

PSCORE: Prints the entire contents of the SCORE array. (See section entitled WORKING DATA ARRAYS AND VARIABLES.)

REVSUM: Prints a summary of what can be accomplished on the most recently evaluated revolution. If IDEVIC="PRNTR", an elaboration of the SCORE matrix is printed. For each FO which has a non-zero unconstrained weight for the revolution, the following is printed: the proposal number (PROP), the FO number (FO), theta at the start (TSTRT) and end (TEND) of the pass over the applicable properly lighted ground site, the total possible weight of the FO (TWEIGH), the unconstrained (WGT U) and constrained (WGT C) weight accomplished on the rev, the mission category of the FO (MCATEG), and the weight accomplished (accepted) so far (CURWGT). Then, regardless of IDEVIC, a summary of the contents of the SCORE array, as limited by the selected Z local vertical pass theta limits, is printed. This data is summarized by mission categories and totals and includes the constrained and unconstrained weight accomplished, the accumulated weight, the total weight possible, and the number of FO's with non-zero unconstrained weights.

STOP or HALT

This option terminates the current session. If IERSAV>0, the accomplishments, as of the end of the current session are written on logical unit "IERSAV" and the program so states before terminating. If IERSAV=0, the program tells the user that "PROPOSAL ACCOMPLISHMENTS WILL NOT BE SAVED OK?". An affirmative response simply terminates the program. A negative response causes the program to request the number of the logical unit on which to write the data.

WARNING - Either way of writing on unit IERSAV involves a rewind operation before writing. Anything previously stored on that unit will be overwritten. NOTE: The IERSAV file (as well as the IEROLD file) must be catalogued and a number assigned (via an @USE card) prior to the beginning of program execution. There is no way to save the data if a file has not been assigned.

BASIC ASSUMPTIONS AND ALGORITHMS

In order to effectively use the program, the mission planner should understand some of the basic assumptions and algorithms used in the program.

The Omega-Theta Transformation

While the basic ground site data is stored in terms of the latitude and longitude of the site corner points, it is transformed to omega-theta space for use by the ground site coverage routine FLYOVR. This omega-theta mapping describes points on earth in terms of the parameters of the orbit passing over them. Omega is the geographic longitude of the ascending equatorial crossing of the orbit which passes directly over the ground site. Theta is the angle, measured in the instantaneous orbit plane, from the ascending equatorial crossing to the point where the spacecraft is directly over the ground site. The spacecraft is said to be on the ascending leg of the orbit when its velocity vector has a northward component. Consequently, $-90 \le \theta \le 90$ degrees on the ascending leg and $90 \le \theta < 270$ degrees on the descending leg. If the absolute value of the latitude of the ground site corner point is less than the inclination of the orbit plane, the point will map into two omega-theta pairs - one corresponding to the orbit which passes over the point on the ascending leg, the other pair corresponds to a descending pass. If the absolute value of the latitude is equal to the inclination, a single omega-theta pair is obtained. If the absolute value of the latitude is greater than the inclination, no values of omega-theta result since the orbit does not pass over the point.

The purpose of using the transformation is to simplify and speed up the computations made in subroutine FLYOVR. Knowing the extremes of omega for the corner points of a site and the omega of a particular orbit of interest for example, one can immediately determine whether or not the orbit passes over the ground site.

The use of the transformation does have one rather awkward implication. Sites which lie just outside the latitude boundaries imposed by the orbit inclination, but which may still be viewable within the field of view of the instruments are not properly handled. Special logic, (not currently in FLYOVR), would have to be added to detect when such sites are viewable. Sites which straddle the critical latitudes are effectively handled in FLYOVR. A simple, but probably adequate, artifice for properly handling sites outside the latitude boundaries, but which should be in the field of view when the spacecraft is at its latitude extremes, would be to redefine the ground site corner point latitudes so that the sites do straddle the critical latitude.

The equations used to determine omega-theta from latitude-longitude are given below.

For $\dot{L}>0$

$$\Theta = \sin^{-1} \frac{\sin L}{\sin i}$$

$$\Omega = \lambda - \tan^{-1} (\cos i \tan \Theta) + R\Theta$$

For $\dot{L}<0$

$$\Theta = 180 - \sin^{-1} \frac{\sin L}{\sin i}$$

$$\Omega = \lambda - \tan^{-1} (\cos i \tan \Theta) + R\Theta + 180$$

where L = geocentric latitude of the point

λ = geocentric longitude of the point

i = orbit inclination

and R = the orbit precession of the longitude of the ascending equatorial crossing (earth rate + inertial precession) in terms of degrees precession per degree travel of the spacecraft.

These equations assume a circular, fixed inclination, fixed period orbit. The orbit inclination and precession terms could be made input variables but they currently are fixed in the program. The equations would have to be changed for a substantially non circular orbit. (The site data currently used is actually expressed in terms of geodetic latitude though the above equations assume geocentric latitude. This anomaly should probably be corrected.)

The basic program does not depend on working in terms of omega-theta space but subroutine FLYOVR would have to be rewritten in order to change. The transformation is also quite useful when working with ground tracks graphically. Figures 1 and 2 depict the world and the United States in omega-theta coordinates for the nominal Skylab orbit. In this projection, ground tracks are represented by vertical straight lines. (These figures will be helpful when one wishes to limit the range of omega considered in the SCAN option or study the geographic areas overflown for a particular revolution.)

Revolution Definition

A revolution is defined as starting when the spacecraft is at the southernmost point in its orbit (θ= -90 degrees) and ending the next time it is at the southernmost point of the orbit (θ=270 degrees). This latitude to latitude (essentially equivalent to a nodal orbit) definition was selected in favor of the longitude to longitude definition commonly used by MSC because it features a constant period revolution. Some computations were simplified this way. The "strange" beginning and end points were selected because there were no sites near 50 degrees south latitude. Thus, the awkwardness of having a contiguous pass occur on two revolution numbers was avoided. A small portion of South America does extend south of 50°S but currently, at least, no sites are defined in that region.

Groundsite Coverage - Overflight Rules

The algorithms used to decide whether the site required by a given FO will be covered on a given revolution require explanation. The logic discussed here is contained in Subroutine FLYOVR. Associated with each FO is a list of the instruments which should be turned on to provide the required data. The field of view of the required instrument having the narrowest field of view is used in the coverage computations.

The range of omega is determined for the required ground site. If the omega of the particular revolution falls within the site range of omega, the pass is immediately deemed acceptable in terms of adequate coverage. If it does not, additional computations are made to determine the adequacy of coverage. First the extent of the field of view in terms of omega is computed. Consider the two points at either end of the edge of the field of view, (along a line orthogonal to the velocity vector) when the spacecraft is positioned at a value of theta equal to the middle of the theta range of the site. The omega values of these points are called the extent of the field of view in terms of omega. Figure 3 shows the omega range of the field of view of the S190 equipment as a function of theta. If the omega range of the site is larger than the field of view, the site is called a large site. The coverage algorithm requires that the spacecraft fly directly over a large site in order for the pass to be counted. If however, the site is small, at least 1/4 of the site must be within the field of view for the pass to count.

If the pass goes directly over the site, the theta limits of the pass are the values of theta when the subvehicle point enters and leaves the site. If an acceptable pass does not go directly over the site and the site extends beyond the edge of the field of view, the theta limits are defined as the values of theta when the subvehicle point of a pseudo-spacecraft flying on an orbit with an omega equal to the omega of the edge of the field of view enters and leaves the ground site. If an acceptable pass does not go directly over the site and the site will be entirely in view, the pass theta limits are equal to the theta limits of the site.

Ground Site Coverage - Lighting Rules

The logic described here is contained in Subroutine PASBAD. Consider the extent of the site measured in terms of the range of orbit arc. At least "ARCMIN" degrees of the overflight must be adequately lit or at least 1/2 the theta range of the pass must be adequately lit. If either of these criteria are satisfied, the pass is acceptable. If necessary, the theta limits of the pass are adjusted so that they only cover the range of the adequately lit ground site.

Ground Site Coverage - Z Local Vertical Pass Limiting

The subroutine PASBAD logic is reapplied to each FO when considering what is accomplished by a Z local vertical pass. This logic attempts to ensure that coverage is not counted if the selected theta limits for the pass cut off too much of the coverage. This checking is done in several print routines, and under the MAXWGT and ACCEPT options.

Constrained vs. Unconstrained Weighting

The unconstrained weight accomplished on a pass is the portion of the FO weight accomplished without regard to what has already been accomplished. If N passes are required to satisfy an FO and W is the total FO weight, a single acceptable pass will be given an unconstrained weight equal to W/N.

The constrained weight accomplished on a pass takes into account what has been accomplished on previously accepted revolutions. If W_{TP} is the total weight possible for the FO, W_U is the unconstrained weight accomplished on a pass, and W_A is the previously accumulated FO weight, the constrained weight is equal to either W_U or W_{TP}-W_A, whichever is least.

The MAXWGT (and SCAN) option optimizes the placement of the Z local vertical pass on the revolution on the basis of the constrained weight accomplished.

Mission Categories

The FO's have been split into categories according to the mission(s) on which they can be satisfied (seasonal constraints): 2 only, 3 only, 4 only, 2 or 3,2,or 4, 3 or 4,2 or 3 or 4, and none. While numerous print options display data broken down into these categories, the only processing effect of the categorization occurs in the MAXWGT option (and therefore SCAN). Maximization will be done considering only those FO's with category numbers included in the value of "MCATEG". For example, if MCATEG=23, only the weights accumulated from FO's in categories 2,3, or 23 are considered. If MCATEG=234, the accomplished weights for every FO are considered.

Repeat Coverage

The following logic applies to the determination of the acceptability of a pass for an FO requiring repeat coverage. If no pass has previously been accepted which covers the FO, any otherwise acceptable pass will be counted. Once at least one pass has been accepted, the dates of all subsequent otherwise acceptable passes will be compared with the already accepted passes to see that they have acceptable date spacing. The algorithm is designed such that the longer the required interval between passes, the more tolerance is allowed the acceptable spacing. Basically, a 10% slop is permitted on the spacing. If for example, 5 day spacing is required, the interval between passes must be an integer multiple of 5 days ± .5 day. If 20 day spacing is allowed, the interval between passes is permitted to be an integer multiple of 20 days ± 2 days.

There does exist a flaw in the pass spacing logic which future users may wish to fix. The logic permits a subsequent pass so long as it occurs on a date which is an even multiple of the required day spacing away from a previously accepted pass. For example, if the passes are required 5 days apart, a pass 20 days later than an already accepted pass would be acceptable. If the proposer has requested 4 passes, 5 days apart, this is acceptable, but if he requested 2 passes 5 days apart it probably isn't. Additional logic incorporating the number of passes required might be desirable. This logic would be added to Subroutine FOSCAN.

WORKING DATA ARRAYS AND VARIABLES

Basic working data is stored in a number of arrays and variables and is available to the various subroutines via labeled common. The contents, source, and use of the arrays and variables will be described here. The names in brackets indicate the labeled common section in which the variables will be found. The labeled common sections are actually defined in PDP elements which are added to the subroutines as required by the Fortran INCLUDE statement.

EREP [EREPC]

The EREP array is essentially a copy of the EREPD file except that the entries for each FO have been reassigned so that all FO's which are in the same mission category, 2, 23, etc., are contiguously represented in EREP. The lines representing FO's which cannot be accomplished because the season constraints do not match any of the missions, are not read into EREP. EREP is dimensioned 15 x 1000. The 15 entries for each FO are identical to the corresponding entries in the EREPD file. Once the EREP array is filled by a file read in subroutine COMMON, during program initialization, it is never altered by the program. EREP is used by a majority of the major subroutines.

SITE [EREPC]

Site contains the corner points of each of the ground sites referenced by the various proposals expressed in terms of the latitude and longitude of the points. Although the latitude information supplied by the proposers as well as Bellcomm is actually expressed as geodetic latitude, it is currently treated as though it were geocentric. This minor anomaly can be most efficiently cured by either converting the latitude information on the file to geocentric or by converting it at the time it is read in subroutine COMMON. (The latter approach would leave the file data directly compatible with most maps.) SITE is dimensioned 2 x 5000 and is used by FLYOVR.

TRAJ [EREPC]

TRAJ contains the basic trajectory information required. Only the data required for the mission currently being studied is maintained in TRAJ. When the variable MISION is changed, COMMON refills TRAJ with the appropriate data. TRAJ is dimensioned 5 x 700. Row 1 contains the revolution number; row 2 contains the geographic longitude of the ascending equatorial crossing; row 3 contains the value of Beta, (the angle between a vector pointing at the sun and the orbit plane at the time of the crossing); row 4 contains the time, measured in days, from 0000 GMT January 1; and row 5 contains the value of theta at orbit noon. The external routine EREP.MISION reads the basic trajectory file, separates it into the three chunks required by the program and excludes those revolutions on which the magnitude of Beta is excessive (greater than 50 degrees currently). TRAJ is used by MAIN, WGTADD, and FOSCAN.

SCORE [SCOREC]

The SCORE array stores a summary of the FO's which can be satisfied on the most recently evaluated revolution. For each FO which can be satisfied, six parameters are recorded.

IEROW - The column number of the FO within the EREP array.

TSTART - The value of theta at the start of coverage.

TEND - The value of theta at the end of coverage.

PWGHU - The unconstrained partial weight accomplished on this revolution - unconstrained means without regard for any previous accomplishments on this FO.

PWGHC - The constrained partial weight accomplished on this revolution - the constrained weight takes into consideration what has previously been done on this FO and consequently represents the value of new accomplishments.

MCATEG - The mission category of this FO, for example, MCATEG = 2 if this FO can be satisfied only during the SL-2 mission. MCATEG can equal 2,3, 4,23,24,34, or 234.

SCORE is written in subroutine FOSCAN, and is used by numerous print routines and the WGTADD routine. The parameter NSCORE (also carried in SCOREC) contains the number of valid columns in SCORE for the current revolution. SCORE is currently dimensioned 6 x 100. If more than 100 FO's can be satisfied on the revolution, a printed alarm message will occur and only 100 columns will be filled.

<u>TOTSCR</u> [SCOREC]

The TOTSCR array is dimensioned 1 x 1000. Each column corresponds to the FO listed in the same column number of the EREP array. The entries in TOTSCR represent the partial FO score accumulated under the ACCEPT option so far. Values in TOTSCR can range from zero to the total possible score for the FO. TOTSCR is written by subroutine WGTADD and is used by numerous print routines and by FOSCAN to determine the constrained score achieved by a revolution under consideration.

<u>PASDAT</u> [SCOREC]

PASDAT has as many columns as there are FOs requiring repeating type coverage. For each column, the first row contains the column number of the applicable FO in the EREP array, the second row contains the total number of passes which have been accepted and which contributed to the coverage requirements of the FO. The remaining rows list the dates on which those accepted passes were made. This data is used by FOSCAN to determine whether a revolution under consideration, which otherwise satisfies the FO, is properly separated in time from previously accepted revolutions. PASDAT is written in subroutine WGTADD.

OMECOV [SCOREC]

The OMECOV array currently exists but is not used. Its intended use is similar to that of PASDAT except that it is associated with the FOs which require total area coverage. The geographic longitude of the ascending equatorial crossing of accepted revolutions would be stored in place of the pass date. OMECOV would be written in WGTADD and used by FOSCAN (or, perhaps FLYOVR) to determine whether a new pass accomplished new coverage.

PASS [SCOREC]

The PASS array stores a summary of the local vertical passes which have been accepted so far. PASS is dimensioned 7 x 200, so up to 200 passes may be accepted before overflow occurs. The data stored in each row is as follows:

1. IREV - The revolution number of the pass.
2. OMEGA - The geographic longitude of the ascending equatorial crossing of the revolution.
3. DATE - The time of the start of the pass measured in days from 0000 January 1.
4. THSTRT - Theta at the start of the pass (in degrees).
5. THEND - Theta at the end of the pass (in degrees).
6. ELS - Sun elevation angle at the start of the pass (in degrees).
7. ELE - Sun elevation angle at the end of the pass (in degrees).

PASS is written in WGTADD and is used by various print routines.

INDEX [EREPC]

INDEX contains pointing information to locate the corner points of a site in the SITE array. INDEX is dimensioned 2 x 600, each column containing the pointing data for the corresponding site number. The first row contains the SITE array column number of the first corner point of the site and the second column contains the number of corners describing the site. INDEX is written by the external routine EREP.SITES, is brought in with the other input data files, and is used by subroutine FLYOVR.

IERPNT [EREPC]

IERPNT contains information which points to the location limits of proposal data in the EREP array for each of the mission categories. IERPNT is dimensioned 7 x 2. Each row contains data pertaining to the mission categories, 2, 23, 24, 234, 3, 34, 4, respectively. The first column lists the EREP column number of the first FO in the mission category, the second column lists the column number of the last FO. IERPNT is written by the external data preparation routine MSORT, is brought in with the EREP file, and is used by COMMON and various print routines.

IEREP [CNSTNT]

IEREP is essentially identical to IERPNT except that it contains pointing information for only those FO's which are to be considered for the current mission. IEREP is dimensioned 4 x 2, corresponding to the four mission categories considered for a given mission. IEREP is written in COMMON and used by FOSCAN and various print routines.

MCAT [CNSTNT]

MCAT is dimensioned 4 x 1 and contains the mission category values applicable for the currently considered mission. If MISION = 2 for example, the four entries in MCAT would be 2, 23, 24, and 234. MCAT is written in COMMON and is used by various print routines.

PRLIST [CNSTNT]

PRLIST is a list of the proposal numbers which contain FO's which can be satisfied on the currently considered mission. It is dimensioned 200 x 1, is written in COMMON, and is used by various print routines.

VARIABLES IN COMMON SECTION [CONTRL]

These variables can all be easily changed either at the time of program initiation or via the CHANGE option. Most of them are assigned nominal values by Subroutine COMMON which hold unless they are changed by the user. Variables ending in DE indicate default variables. A similarly named variable will be set to the value of the default variable unless the user specifies a different value at the time it is to be used.

NAME	DESCRIPTION	NOMINAL VALUE
APASDE	The default value of ARCPAS which is the arc length of a Z-local vertical pass under the MAXWGT and SCAN options.	60 degrees
ARCMIN	The spacecraft must be over an adequately lighted site for at least ARCMIN degrees for the pass to be counted unless the site is less than 2*ARCMIN long in terms of theta.	1.0 degree
IDEVIC	Identifies the user's output device and controls the volume of print under certain print options. Can be equal to 'PRNTR' or 'TERMNL'.	TERMNL
IERSAV	= 0, Do not save accomplishments at the end of the computer run, (A last chance to save the data will be offered following entry of the STOP option.) >0, save mission accomplishments on unit "IERSAV" at the end of the computer run.	0

IEROLD = 0, Initialize the current run from scratch 0

>0, Initialize the current run as of the end of a previously saved run. Necessary data is stored on logical unit "IEROLD." IEROLD is only meaningfully non-zero at program initiation. Changing the value of IEROLD via the CHANGE option will be ignored.

MCATDE The default value of MCATEG. Only FO's which are in mission categories included in MCATEG will be considered in optimizing the location of the Z-local vertical pass under the MAXWGT or SCAN options. If MCATEG = 23, for example, only FO's in categories 2, 3, or 23 will be considered. 234

MISION Indicates the mission currently being studied. Can be set to "2" or "3" or "4". 2

PERWT Scores accomplished on an accepted pass will be multiplied by PERWT before being added to TOTSCR. 1.

THMADE The default value of THMAX. THMAX is the maximum value of theta which will be considered in evaluating a revolution under option EVALUATE, MAXWGT, or SCAN. 270.

THMIDE The default value of THMIN. THMIN is the minimum value of theta which will be considered in evaluating a revolution under options EVALUATE, MAXWGT, or SCAN. -90.

VARIABLES IN COMMON SECTION [EREPC]

NAME DESCRIPTION

ITOTRJ The number of revolutions on the trajectory file for the currently considered mission. This parameter is read in with the trajectory data and is computed and written by the auxiliary data routine

MISION. The number of revolutions on which Beta is too large are not included in the ITOTRJ count. Used in FOSCAN.

NBETA1 The revolution number of the first revolution excluded from the mission by high beta. Also written by MISION, and read in with the trajectory data.

NBETA2 The revolution number of the last revolution excluded from the mission by high beta. Also written by MISION, and read in with the trajectory data (TRAJ). Used in FOSCAN.

LINE The column number of the trajectory data pertaining to a revolution being evaluated in the TRAJ array. Computed and used in FOSCAN.

VARIABLES IN COMMON SECTION [SCOREC]

NAME DESCRIPTION

NSCORE The number of valid entries in the score matrix, equal to the number of FO's which could be covered on the most recently evaluated revolution regardless of prior accomplishments.

IPASS The number of passes which have been accepted so far--indicates the number of valid columns in the PASS array. Written by WGTADD.

IDAY The integer value of the day of the year on which the current revolution occurs. Strictly speaking, IDAY is a local variable in FOSCAN which probably should not be in Common.

VARIABLES IN COMMON SECTION [CNSTNT]

DTOR* The multiplicative conversion factor from degrees to radians. Set to .01745329 in COMMON.

RTOD The multiplicative conversion factor from radians to degrees. Set to 1/DTOR in COMMON.

XINCL* The orbit inclination expressed in degrees. Set to 50. in COMMON.

* These variables are not set or included in common section [CNSTNT] in the attached listings - See the section entitled KNOWN ANOMALIES AND INCONSISTENCIES.

PI* Set to 3.14159265 IN COMMON.

EARRAT* The rotation rate of the earth in degrees per hour. Set to 15.04104953 in COMMON.

PRERAT* The inertial precession rate of the orbit expressed in degrees per hour. Set to - .21174988 in COMMON.

NUMPRO The number of proposals which have FO's which can be accomplished in the currently considered mission = number of valid entries in PRLIST. Computed in COMMON and used in some print routines.

ORBPER The orbit nodal period expressed in minutes. Set to 93.283019 in COMMON.

HAS190 The half angle field of view of the S190 cameras. Set to 10.6 degrees in COMMON. Used in LONRAN and FLYOVR.

HAS191 The half angle field of view of the S191 equipment. Set to 20.0 degrees in COMMON. Used in LONRAN and FLYOVR.

HAS192 The half angle field of view of the S192 equipment. Set to 5.0 degrees in COMMON. Used in LONRAN and FLYOVR.

HAS193 The half angle field of view of the S193 equipment. Set to 12.4 degrees in COMMON. Used in LONRAN and FLYOVR.

HAS194 The half angle field of view of the S194 equipment. Set to 7.5 degrees in COMMON. Used in LONRAN and FLYOVR.

RE* The radius of the earth expressed in feet. Set to 20925738.2 in COMMON. Used by LONRAN.

RORB* The radius of the orbit expressed in feet. Set to 22353625.3 in COMMON. Used by LONRAN.

VARIABLES IN COMMON SECTION [PDATA]

IREV The revolution number to be evaluated or most recently evaluated. This is supplied by the user when requesting the EVALUATE option or by the program when the SCAN option has been requested. Used by FOSCAN, WGTADD, and various print routines.

THSTRT The value of theta at the start of a Z-local vertical pass expressed in degrees. THSTRT is automatically written by WGTMAX in the MAXWGT option but may be changed by the user for print options and at the time of accepting a pass.

THEND Similar to THSTRT except it is the value of theta at the end of the pass.

OTHER VARIABLES COMMONLY SEEN BY THE USER

IFDAY The first day to be considered in the SCAN option where day 0 corresponds to 0000 to 2400 GMT on January 1. Supplied by the user.

ILDAY The last day to be considered in the SCAN option. Definition is otherwise the same as IFDAY.

WGTMXC The maximum new FO weight which can be accomplished on a revolution during a pass of ARCPAS degrees taking into account the weight already accomplished. WGTMXC is computed and displayed by WGTMAX following completion of the MAXWGT option.

WGTMXU The unconstrained weight which could be accomplished assuming the same location and length of pass which produced WGTMXC. Also computed and displayed by WGTMAX. NOTE: WGTMXC is maximized with the MAXWGT option, not necessarily WGTMXU.

PROGRAM AVAILABILITY

Three copies of the program currently exist. One is on saved files at Bell Laboratories' Murray Hill New Jersey location. This copy is associated with the newly installed Univac 1106 serving the Cost Systems Studies Center. A second copy, written on magnetic tape has been delivered to the Mission Planning and Analysis Division at the Manned Spacecraft Center, Houston, Texas. Mr. Kenneth Young is the point of contact. The third copy, also on tape, has been delivered to Mr. Don M. Turner, Space Station Task Force, NASA Headquarters, Washington, D.C. Each copy includes relocatable and symbolic versions of the necessary routines, including the auxiliary data set-up routines, and all necessary input data files. Instructions for reading the necessary information from the tapes are included in the section entitled PROGRAM USER'S GUIDE. A copy of the crude flow charts from which several of the subroutines were written has been delivered to Mr. Max Kilbourn of the Mission Planning and Analysis Division at MSC.

PROGRAM USER'S GUIDE

This User's Guide will assume the user has one of the two copies of the tape available. The version stored at the BTL Murray Hill, UNIVAC 1106 has some of the data and programs on differently named files than is assumed here. The tape version is more compact in its use of files.

Instructions for Reading the Program Tape onto Fastrand Files

The fastrand files must initially be assigned after the RUN card has been submitted. There are three ways to assign the files:

(1) @ASG,CP FILENAME.where FILENAME is the name of the file. This assign statement causes FILENAME to be catalogued if the run terminates normally (C), as a public (P) file which subsequently can be referenced in any run by using the same project field on the run card as is currently being used.

(2) @ASG,T FILENAME. This specifies that the file is to be temporary. It will exist only from the time it is assigned until run termination.

(3) @ASG,AX FILENAME. This specifies that the file is currently catalogued (A) and that this run will have exclusive use (X) of the file until this run is terminated. This is the type of assign statement to use to assign a file catalogued during a previous run.

The required file names to be assigned are EREP, EREPD, SL2DAT, SL3DAT, SL4DAT, NEWANO, and ERDATA. Each must be assigned in an individual assign statement.

The format of control cards to be submitted to read the program tape into files is given below:

```
@RUN    RUN-ID,ACCOUNTING,PROJECT,RUNNING TIME, PAGES
@ASG,CP(or AX or T)     EREP
@ASG,CP(or AX or T)     EREPD
@ASG,CP(or AX or T)     SL2DAT
@ASG,CP(or AX or T)     SL3DAT
@ASG,CP(or AX or T)     SL4DAT
@ASG,CP(or AX or T)     NEWANO
@ASG,CP(or AX or T)     ERDATA
@ASG,TM     A, T, XXXX  (where XXXX is the tape number,
                        T indicates a temporary assign-
                        ment, and M specifies that the
                        tape is to be read in medium
                        density, 556 bits per inch)

@COPIN      A.,EREP.  (This copies file 1 of the tape
            into program file EREP.  It contains the
            Earth Resources Experiment Analysis Program
            in the form of 33 symbolic, 27 relocatable
            and 1 absolute elements.  The element names
            can be listed by @PRT,T EREP).

@COPIN      A.,EREP.  (File 2 of the tape.  This adds to
            the EREP file the card-image row data ele-
            ment NEWDAT containing the array of site num-
            bers and the corresponding array of corner
            points.)

@COPIN      A.,EREPD.  (File 3 of the tape.  This copies
            the card-image raw proposal information
            matrix into EREPD.EREPD)

@COPY,G     A.,SL2DAT.  (File 4 of the tape.  This copies
            into SL2DAT, the trajectory information for
            SL-2.  The data is already formatted to be
            read by the main program)

@COPY,G     A.,SL3DAT.  (File 5 of the tape.  This copies
            into SL3DAT the trajectory information for
            SL-3.  The data is already formatted to be
            read by the main program)
```

```
@COPY,G      A.,SL4DAT.  (File 6 of the tape.  This copies
             into SL4DAT the trajectory information for
             SL-4.  The data is already formatted to be
             read by the main program)

@COPY,G      A.NEWANO.  (File 7 of the tape.  This copies
             into data file NEWANO, the Bellcomm generated
             trajectory data from which SL2DAT, SL3DAT,
             and SL4DAT are created)

@COPY,G      A.,ERDATA.  (File 8 of the tape.  This
             copies into data file ERDATA the proposal
             matrix sorted by FO's into mission cate-
             gories and formatted to be read by the EREP
             program)
```

One additional file, LATLON, is required which was not included on the tape. This file, however, can be created from other data in the tape. In addition to writing the file, the following control cards will yield a printout describing the groundsites in both latitude/longitude coordinate and omega/theta coordinates.

```
  @ASG,CP (or AX or T)    LATLON
**@ASG,T      THOMEG
  @USE        2, LATLON
**@USE        3, THOMEG
  @ASG,AX     EREP
  @FOR        EREP.SITES, SITES
  @XQT
  @ADD        EREP.NEWDAT
```

After the necessary files have been created from the tape, the runs described below may be executed.

2. <u>Executable Runs</u>

A. To execute the main program:

```
 @RUN     RUN-ID,ACCOUNTING,PROJECT,RUNNINGTIME,PAGES
 @ASG,AX  EREP
*@ASG,AX  SL2DAT
*@ASG,AX  SL3DAT
*@ASG,AX  SL4DAT
```

** The statement WRITE (3,OUT2) can be safely removed from the SITES program since the file THOMEG is no longer used by the main program. If it is removed, the indicated cards can be deleted.

```
*@ASG,AX  ERDATA
*@ASG,AX  LATLON
*@ASG,T   TFILE     (This is a temporary file for
                    scratch writing)
*@USE      2,SL2DAT
*@USE      3,SL3DAT
*@USE      4,SL4DAT
*@USE     10,ERDATA
*@USE     11,LATLON
*@USE     15,TFILE
 @ASG,AX     MSC1   (These two files may be catalogued
 @ASG,AX     MSC2   and assigned by the user in order
 @USE     20,MSC1   to save the current mission status
 @USE     21,MSC2   at the end of an execution or to
                    initialize a run from the mission
                    status recorded during a previous
                    execution.
 @XQT        EREP.ERMAP
```

B. To list the total weight assigned each proposal:

```
@RUN      RUN-ID,ACCOUNTING,PROJECT,RUNNINGTIME,PAGES
@ASG,AX   EREP
@ASG,AX   EREPD
@FOR      EREP.TOWGT,TOWGT
@XQT
@ADD      EREPD.EREPD
```

C. To create and list trajectory files for SL-2, SL-3, SL-4 from a Bellcomm - generated ephemeris file:

```
@RUN      RUN-ID,ACCOUNTING,PROJECT,RUNNINGTIME,PAGES
@ASG,AX   EREP
@ASG,AX   SL2DAT
@ASG,AX   SL3DAT
@ASG,AX   SL4DAT
@USE      2,SL2DAT
@USE      3,SL3DAT
@USE      4,SL4DAT
@ASG,AX   NEWANO
@USE      9,NEWANO
@FOR      EREP.MISION,MISION
@XQT
```

* For the sake of convenience, one command
@ ADD EREP.FILES
may be typed in place of the indicated commands. EREP.FILES is a data element containing all of the indicated commands and, when added, causes the commands to be executed.

D. To sort and list FO's by mission category:

```
@RUN      RUN-ID,ACCOUNTING,PROJECT,RUNNINGTIME,PAGES
@ASG,AX   EREP
@ASG,AX   ERDATA
@ASG,AX   EREPD
@USE      10,ERDATA
@FOR      EREP.MSORT,MSORT
@XQT
@ADD      EREPD.EREPD
```

3. Instructions for changing data files

A. Adding or changing proposal data

a. Modify element EREPD located on file EREPD

b. Execute a conversion run:

```
@RUN      RUN-ID,ACCOUNTING,PROJECT,RUNNINGTIME,PAGES
@ASG,AX   EREP.
@ASG,AX   ERDATA
@ASG,AX   EREPD
@USE      10,ERDATA
@FOR      EREP.MSORT,MSORT
@XQT
@ADD      EREPD.EREPD
```

B. Adding or changing site data

a. Modify element NEWDAT located on file EREP

b. Execute a conversion run:

```
  @RUN      RUN-ID,ACCOUNTING,PROJECT,RUNNINGTIME,
            PAGES
  @ASG,AX   LATLON
**@ASG,T    THOMEG
  @USE      2,LATLON
**@USE      3,THOMEG
  @ASG,AX   EREP
  @FOR      EREP.SITES,SITES
  @XQT
  @ADD      EREP.NEWDAT
```

** See previous footnote concerning file THOMEG.

C. Adding or changing trajectory data

There is no card-image file containing trajectory data. Therefore, modifications can be made only by regenerating or reprocessing the data. In either case, it must appear in final form on the trajectory file (i.e. SL2DAT, SL3DAT, or SL4DAT) as one binary record containing:

a. The integer number of revolutions included on the file

b. The integer number of the first rev with the absolute value of beta greater than the maximum value allowed.
(If the absolute value of beta is never greater than the maximum value allowed, set this number and (c) to zero).

c. The integer number of the last rev with the absolute value of beta greater than the maximum allowed. See (b).

d. A floating point array dimensioned 5x700 where, for the Kth entry

(1,K)	is the rev number
(2,K)	is the geographic longitude (degrees) of the ascending equatorial crossing
(3,K)	is the value of beta (the angle between the sun vector and the orbit plane)
(4,K)	is the time of the ascending equatorial crossing expressed in days from 0000 GMT on January 1
(5,K)	is the value of beta at orbit noon

D. Changing mission dates and running with the Bellcomm - supplied trajectory data.

In element MISION of file EREP, modify as required the values of SL2STR, SL2END, SL3STR, SL3END, SL4STR, SL4END to the desired start (suffix STR) and end (suffix END) times (in days from zero time on January 1). If the maximum allowed magnitude of beta is to be different than 50°, the statement: "IF (ABS(BETSUN).LT.50) GO to 60" should be changed to reflect the new limit.

b. Execute a conversion run:

```
@RUN      RUN-ID,ACCOUNTING,PROJECT,RUNNINGTIME,
          PAGES
@ASG,AX   SL2DAT
@ASG,AX   SL3DAT
@ASG,AX   SL4DAT
@USE      2,SL3DAT
@USE      3,SL3DAT
@USE      4,SL4DAT
@ASG,AX   NEWANO
@USE      9,NEWANO
@FOR      EREP.MISION,MISION
@XQT
```

4. Brief Dictionary of Terms and Variables Required by a User

VARIABLE	DESCRIPTION	NOMINAL VALUE
APASDE	Central Arc of ZLV passes	60.
ARCMIN	Number of degrees of site which must be within spacecraft field of view in order to be considered an acceptable contact	1.0
IDEVIC	User output device. Controls volume of printout = TERMNL or PRNTR	TERMNL
IERSAV	= 0, Do not save mission status at end of computer run > 0, Save mission status on unit "IERSAV" at end of computer run	0

IEROLD	= 0, Initialize current run from scratch	0
	> 0, initialize current run at previously generated mission status read from unit "IEROLD"	
IFDAY	First day to be scanned for best revs in terms of FO weights accomplished. Day 0 begins at 0 hours, January 1. Required input for the scan option	
ILDAY	Last day to be scanned for best revs. See IFDAY	
FO	Flight objective. Each proposal is broken down into distinct objectives. Each objective is given a weight so that the weight of all FO's of a proposal = 100	
MCATDE	Mission category with respect to which a pass is optimized	234
MISION	Mission being studied	2
PERWT	Weighting factor to be applied to FO weights achieved during a pass	1.0
THEND	Value of theta at the end of the optimum arc of a rev. Calculated by the MAXWGT option. Value can be changed by user	
THMADE	Largest value of theta to consider in optimizing a pass or evaluating a revolution	270.
THMIDE	Smallest value of theta to consider in optimizing a pass or evaluating a revolution	-90.
THSTRT	Value of theta at the beginning of the optimum arc of a rev. See THEND	
WGTMXC	Maximum weight which can be accomplished by a rev during a pass of APASDE degrees,	

taking into account work already accomplished on the FO's under consideration

WGTMXU Maximum weight which can be accomplished by a rev during a pass of APASDE degrees, not taking into account work already accomplished on the FO's under consideration

5. Brief Description of the Available Options (and Spelling)

Evaluate or Eval or E

Evaluate a designated revolution in terms of which, and how many, of the EREP FO's can be accomplished. Put the results in the SCORE array.

MAXWGT OR M

Determine the range of arc which maximizes the F.O. weights accomplished by a rev.

SCAN OR S

SCAN every rev from IFDAY to ILDAY (inclusive) within specified OMEGA limits, if any, and select the N best revs in terms of the maximum F.O. weights accomplished.

ACCEPT OR A

Cause the program to simulate performing the EREP pass. Update current status information to account for accomplishments.

PRINT OR P

According to which suboption is submitted, print information summary of one of the following:

- Revolution
- Proposal
- Current Mission Status

See following page for spelling and description of the print suboptions.

CHANGE OR C

Change the value of one or more control variables such as the mission or the default value of the length of the arc of ZLV passes.

STOP OR HALT

Save current mission status if specified, and exit from program.

6. PRINT SUBOPTIONS

OPTION	DESCRIPTION
ALDONE	List the FO's which have been completely accomplished.
ERESUM	Provide a summary of the conditional, unconditional, and accumulated weights accomplished for every proposal being considered for current mission.
MISSUM	Summarize the current mission status by mission category in terms of the number of FO's worked on and the weights accumulated.
NOWORK	List each FO for which the accumulated weight is zero.
PASSUM	Summarize the pass data accepted so far
PRGSUM	Summarize the accumulated FO weights accomplished and the total number of FO's worked on for each category of each mission.
PRPSUM	Print the conditional, unconditional, and accumulated weights for a given proposal.
PSCORE	Print the score array.
REVSUM	Print a summary of the score array.

SUBROUTINE DESCRIPTIONS

The following pages describe each of the program subroutines. The program entry point or main routine is included under the title MAIN. Appendix A contains the program listings.

Subroutine ALLDON

Purpose: List all FO's relevant to the current mission which have been completely accomplished by accepted passes. Conclude with a summary, by mission category, of completed FO's.

Calling Sequence: Call ALLDON

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT

Calls: CINOTE

Called By: MAIN

Description: ALLDON searches the list of FO's which can be accomplished on the current mission for those which have been completed. When a completed FO is encountered, the following information is printed: proposal number, FO number, mission category, total possible FO weight, and total accumulated weight. When the entire list has been searched, a summary of the above information is printed by mission category. The following data is given: mission category, total possible weight of FO's completed, total accumulated weight of FO's completed, and total number of FO's completed.

Should a console interrupt occur during the first stage of printing, that printing will be terminated. However, the remaining FO's will be scanned for completed entries, and the summary will be given.

Subroutine CINOTE

Description: CINOTE is a UNIVAC 1108 Assembler Language subroutine which allows a user, operating in an interactive mode, to halt undesired printout without terminating program execution. The user does this by depressing a control interrupt key which causes control to switch from the print mode and to return to the next instruction in the subroutine causing the printing. A call to CINOTE must be made in the beginning of each subroutine during which the user might desire to effect an interrupt. The calling sequence is:

Call CINOTE (IPFLAG) where IPFLAG must be set to zero. The value of IPFLAG is changed by an interrupt, and provision is made for resetting it to zero.

Subroutine COMMON

Purpose: Initialize control parameters and arrays, read in data base, set up COMMON sections. Change arrays when MISION is changed.

Calling Sequence: Call COMMON(ICHFLG) where ICHFLG = 0 indicates initial set-up of program, and ICHFLG = 1 indicates user's desire to change the mission under consideration.

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT

Calls: None

Called By: MAIN

Description: Initially, COMMON assigns default values to a group of control variables. It then allows the user to change the value of one or more of these variables using a namelist input. When the variable values have been finalized, COMMON reads in the proposal matrix, the site array, and the trajectory information. It loads the relevant mission category numbers into the MCAT array and sets up pointers to the appropriate blocks of data in the proposal (EREP) matrix. Finally it fills array PRLIST with the proposal numbers for the current mission.

When COMMON is called with the value of ICHFLG = 1, the set of trajectory data for the new mission is read into the TRAJ array, overlaying the data already there. Additionally, the mission category numbers in the MCAT array and the pointers to the EREP matrix are revised. Finally, PRLIST is filled with the proposal numbers for the new mission.

Subroutine ELTHET

Purpose: Given BETA and the value of theta at orbital noon, solve for the values of theta for a given sun elevation angle.

Calling Sequence: Call ELTHET(BETA, TNOON, EL, THETUP, THETDN, NOSUCH) where

BETA is the angle between the solar vector and the orbit plane (degrees). (input)

TNOON is the value of theta at orbit noon (degrees). (input)

EL is the sun elevation (degrees). (input)

THETUP is the value of theta at which the sun elevation is EL degrees and rising. (output)

THETDN is the value of theta at which the sun elevation is EL degrees and falling. (output)

NOSUCH = 0 unless there is no value of theta corresponding to EL degrees sun elevation, in which case it is set to 1. (output)

Calls: SIN, COS, ACOS

Called By: FOSCAN

Description: ELTHET determines the arc length from orbital noon to a given sun elevation. It then subtracts this value from theta noon to yield THETUP and adds it to theta noon to determine THETDN and returns to the calling routine. If no theta values can be found for a given sun elevation, flag NOSUCH is set to 1 and control returns to the calling routine.

Subroutine FLYOVR

Purpose: Determine whether a given revolution passes over, or in view of, a site. If it does, compute theta limits of pass (and, if total coverage is required, the limits of omega effectively covered*).

Calling Sequence: Call FLYOVR (OMEGA, ISITE, FOV, ITOTAL, IOVER, TSTRT, TEND, OMMIN, OMMAX) where

OMEGA is the longitude of the ascending equatorial crossing. (degrees) (input)

ISITE is the site number. (input)

FOV is the half angle field of view of the instrument with the narrowest field of view required by the FO. (degrees) (input)

ITOTAL = 0, total coverage of site not required

= 1, total coverage of site required (input)

IOVER = 0, rev does not pass over site

= 1, rev passes over site on ascending leg

= 2, rev passes over site on descending leg

= 3, rev passes over site both ascending and descending (output)

TSTRT(2) value of theta at start of pass over site. TSTRT(1) contains value of theta for ascending pass. TSTRT(2) contains value

* Not implemented.

of theta for descending pass. (output)

TEND(2) value of theta at end of pass over site. (output)

OMMIN(2) minimum value of omega at which site can be viewed. OMMIN(1) contains value of omega for ascending pass. OMMIN(2) contains value of omega for descending pass. (output)

OMMAX(2) maximum value of omega at which site can be viewed. (output)

Common Sections Used: EREPC, CNSTNT

Calls: ASIN, COS, LONRAN, SIN, SLOPE

Called By: FOSCAN

Subroutine FLYOVR

Description: The corner points of the site being considered are extracted from the SITE matrix and are converted to omega-theta coordinates, ascending and descending. Next the maximum and minimum omega values, both ascending and descending, are determined.

The flow of events described next occurs twice, once for the ascending portion of the orbit and once for the descending portion: The average value of theta of the site is calculated and used in a call to LONRAN to determine the edges of the field of view in terms of omega limits. If the orbit passes directly over the site, that is, if the omega of the revolution falls within the omega range of the site, the pass is considered acceptable, and the theta limits of the pass are computed as the values of theta when the subvehicle point enters and leaves the site. (If the site is shaped such that the subvehicle point enters, leaves, enters again and then leaves again, all in one leg of the orbit, only the first entry and last exit values are used.) If the spacecraft does not pass directly over the site and the site is large--defined as having an omega range greater than the omega range of the field of view--the pass is not counted. If the site is small and at least 1/4 of the omega range falls within the field of view, the pass is counted. The theta range of a small site which straddles the edge of the field of view is computed as the values of theta when the subvehicle point of a pseudo spacecraft on an orbit with omega equal to the edge of the field of view enters and exits the site. If a small site is not directly overflown but lies entirely within the field of view, the largest and smallest values of theta for the site are used for the pass theta limits. The value of IOVER is set according to the coverage acquired as defined in the calling sequence above. After both the ascending and the descending portions of the orbit have been considered, control returns to FOSCAN.

Subroutine FOSCAN

Purpose: Scan the EREP matrix to determine which FO's can be satisfied on a given revolution.

Calling Sequence: Call FOSCAN

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT, PDATA

Calls: CINOTE, ELTHET, FLYOVR, PASBAD, SUNEL, TIMDAT

Called By: MAIN, MASCAN

Description: The trajectory data of the rev being considered is extracted from the TRAJ array, and the date of the rev is converted from days since zero hours, January 1, 1973, to month, day and hours by a call to TIMDAT. Control then enters a loop which scans all of the mission FO's for possible accomplishments by the rev.

First the date of the rev is checked against the months desired by an FO. If there is an agreement, the FO is considered further; if not, control loops to the next FO. Next, the flag ITOTAL is set, indicating either that the FO requires total coverage (ITOTAL = 1) or it does not (ITOTAL = 0).

The sensors desired by the FO are next considered, and the half-angle field of view of the spacecraft is set to the narrowest field of any of its desired sensors. This information, combined with the total coverage indicator and pertinent trajectory data, is used in the call to FLYOVR to determine if the FO site will be acceptably in view.

If the site is not overflown, no further consideration is given the FO. However, if the site is overflown, the subroutine checks whether the lighting conditions at the site meet the FO specifications. If the lightinr conditions reduce the usable ranre of theta of the overflight, the appropriate theta limits are adjusted and subroutine PASBAD is called to determine whether the good part of the pass is lonr enourh to be acceptable. If it is, the program finally looks at the coverage requirements. Since no coding is implemented for total coverage, it is treated as a sinrle pass requirement. For FOs specifyinr repeatinr coverage of the site, the date of the rev is checked against the dates, if any, of previous overflights of the site in order to determine if the pass occurs on an acceptable date spacinr.

Next the constrained weight and the unconstrained weight accomplished by the pass over the site are determined. The unconstrained weight is calculated by dividinr the total weight allotted the FO by the total number of overflights desired. The constrained weirht is determined in the same manner, but, if the resultant weight plus the weight already accomplished is greater than the total possible weight, the constrained weight is set to the difference between the total possible weirht and the total accumulated weight. The minimum constrained value = 0.

If a site meets all of the requirements for an acceptable pass, the following information for the rev is entered in the SCORE array: the line number of the FO in the EREP matrix, theta at the start of the pass over the site, theta at the end of the pass over the site, the unconstrained and constrained weights accomplished by the pass, and the mission category of the FO.

Subroutine LATLON

Purpose: To convert coordinates expressed by latitude and longitude into theta/omega coordinates, with the effects of the earth's rotation and orbit regression rate included in the computation.

Calling Sequence: CALL LATLON (XLATC, XLON, ØMEGA1, ØMEGA2.)

XLATC The latitude coordinate of a point, measured in a plane normal to the equator. (Degrees) (Input)

XLON The longitude coordinate of a point, measured in the plane of the equator with respect to a reference meridian. (Degrees) (Input)

OMEGA1 The longitude of the ascending equatorial crossing of an orbit whose ascending side crosses the point (XLATC, XLON). (Degrees) (Output)

OMEGA2 The longitude of the ascending equatorial crossing of an orbit whose descending side crosses the point (XLATC, XLON). (Degrees) (Output)

Common Sections Used: CNSTNT

Calls: No other subroutine.

Called By: Subroutine LONRAN.

Description: The subroutine operates by first calculating the ascending and descending values of theta corresponding to the calling argument, XLATC; (The values of theta are functions of latitude only). The two values of omega, OMEGA1 and OMEGA2, corresponding to the ascending and descending orbits which cross the point (XLATC, XLON) are then calculated. This involves subtracting two terms from the longitude (XLON) of the given point. The first is a geometric term which is simply the displacement in longitude between the ascending equatorial crossing and a point on the ground track at latitude XLATC; the displacement in longitude is bi-valued, one for the ascending and one for the descending side of the orbit. The second term is the displacement in longitude which ensues (due to earth's rotation and orbit regression) during the time the spacecraft travels through the centrol angle theta, again bi-valued.

Subroutine LONRAN

Purpose: To determine the difference in the omega coordinates between a given point and the points at either edge of the field of view at that point. The edges of the field of view lie on a line orthogonal to the ground track.

Calling Sequence: CALL LONRAN (THETA, FOV, DOMN, DOMP.)

THETA — The angle between the ascending equatorial crossing and a point on the ground track, measured in the instantaneous orbit plane. (Degrees) (Input)

FOV — The half angle field-of-view of an EREP instrument. (Degrees) (Input)

DOMN — The negative difference in omega, with respect to the omega of the ground track. It is equivalent to the difference in omega between the given orbit and an orbit which passes over a point at the extreme edge (measured normal to the flight direction) of the field-of-view. The negative difference corresponds to the point at the left edge of the field-of-view for ascending orbits, and corresponds to the point at the right edge of the field of view for descending orbits. (Degrees) (Output)

DOMP — The positive difference in omega, with respect to the omega of the ground track. The positive difference corresponds to a point at the right edge of the field of view for ascending or-

C.2.

bits, and to points at the left edge of the field of view for descending orbits. (Degrees) (Output)

Common Sections Used: CNSTNT

Calls: Subroutine LATLON.

Called By: Subroutine FLYOVR.

Description: The central angle from the spacecraft to the edge of the field of view is calculated based on the half angle field of view. Next the latitude and longitude of the point at the edges of the point of view are determined. They are computed on the basis of lying on a great circle orthogonal to the spacecraft orbit and intersecting it at the location of the spacecraft. Subroutine LATLON is then called to determine the values of omega of the field of view edge points. If the value of theta for the spacecraft indicates that it is on the ascending leg of the orbit, the ascending omega values for the edge of the field of view are used and vice versa. Finally the difference between the omega values and the reconstucted omega of the orbit are determined and returned to the calling routine.

MAIN Routine

Purpose: Provide the starting point of the program and control the program flow by means of options input by the user.

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT, PDATA, PRNT

Calls: ALLDON, CINOTE, COMMON, FOSCAN, MASCAN, NONDON, PPASS, PPROP, PSCRMC, PTSCRM, PTSCRP, THEMAX, WGTADD

Description: The MAIN routine is the program starting point. Initially it calls subroutine COMMON with the variable ICHFLG in the calling argument set to zero. This allows the current mission to be defined and the corresponding arrays and variables to be initialized.

After the program has been set up, MAIN accepts the first option from the user. As each option is carried out, control returns to this point to read the next option. There are seven options available to the user. After an option is read, control transfers to the section of MAIN appropriate for that option.

Following is a description of the program activity as a result of each option (underlined letters may be typed instead of the formal option name):

EVAL option. This causes a designated revolution to be evaluated in terms of which and how many of the EREP FO's can be accomplished. The user types the number of the rev he wishes evaluated. MAIN determines if the rev is valid for the current mission. If it is valid, MAIN calls FOSCAN to evaluate it. If it is invalid, an appropriate diagnostic is printed.

MAXWGT option. This determines the range of arc which maximizes the FO weights accomplished by a rev. The length of arc to be maximized, the mission category to be maximized, and the start and end values of theta to be considered are all set to the default values APASDE, MCATDE, THMIDE, THMADE, respectively. The user is then allowed to change any or all of the assigned values through input in namelist MAX. MAIN then calls THEMAX to determine the best range of arc.

SCAN option. This option causes a scan of every rev within specified time limits and within specified omega limits, if any, to select the n best revs in terms of maximum FO weights accomplished. The user is asked to type the first and last days during which he desires the revs scanned. Subroutine MASCAN is then called.

ACCEPT option. This causes the program to simulate performing the EREP pass by updating the current mission status information to account for the accomplishments. The user is shown the current values of the rev number and the multiplier PERWT. If they are not what he desires, he may so indicate, in which case control returns to read another option.

PRINT option. According to which suboption is submitted, the PRINT option causes information to be printed about one or more of the following: the current rev, a particular proposal, the current mission status. The available suboptions are explained on another page. MAIN reads the name of the suboption typed in by the user, and calls the appropriate subroutine to effect the desired information gathering and printing.

CHANGE option. This option allows the user to change one or more of the control variables. These include the central arc of a ZLV pass

(APASDE), the number of the mission (MISION), the mission category on which to maximize a pass (MCATDE), the minimum and maximum values of theta to consider for pass optimization (THMIDE, THMADE), the weighting factor to be applied to FO weights achieved during a pass (PERWT), the unit number of the file on which to save the mission status (IERSAV), the unit number of the file from which to read a previous mission status (IEROLD), the output device (IDEVIC), and the number of degrees of a site which must be within the field of view to be considered an acceptable contact (ARCMIN). The program reads the changes through namelist CONTRL. If the mission number is changed by the user, MAIN calls COMMON with argument ICHFLG = 1 in order to acquire the correct trajectory data.

STOP (or HALT) option. This option allows the program to terminate <u>after</u> first having saved the current mission status if desired. If the user has assigned a value to IERSAV, the mission status is automatically written onto the unit and program execution is halted. However, if the user has not specified a value for IERSAV, he is given an opportunity to do so before execution stops.

Subroutine MASCAN

Purpose: Examine every rev occurring from IFDAY through ILDAY and within specified omega limits, if any, determine which accomplish the most in terms of FO weight accomplished.

Calling Sequence: Call MASCAN (IFDAY, ILDAY) where ILDAY is the first day of the scan. January 1 at Greenwich is Day 0. (Input)
ILDAY is the last day of the scan. (Input)

Common Sections Used: CNTROL, EREPC, PDATA

Calls: FOSCAN, THEMAX

Called By: MAIN

Description: First the user is allowed to specify omega windows for the scan about to be executed and is then permitted to change the values of four control variables (APASDE, MCATDE, THMIDE, THMADE) if so desired. The scan for best revs begins with the TRAJ matrix being searched for all revs occurring within the date and omega constraints. Each of the acceptable revs is evaluated and maximized by a call to FOSCAN and to THEMAX. The results are stored in an array, and, after all possible revs in the trajectory matrix have been considered, the array is scanned for the N entries exhibiting the highest constrained weights. These entries are printed for the user.

Limitation - an array (FLY) in MASCAN is currently sized to allow up to 700 revolutions to be considered. If more are contained between IFDAY and ILDAY, an error will most probably occur. This possible, but unlikely, event is not alarmed.

Subroutine NONDON

Purpose: List the FO's which can be done on the current mission for which the accumulated weight equals zero.

Calling Sequence: Call NONDON

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT

Calls: CINOTE

Called By: MAIN

Description: NONDON searches the accumulated score array (TOTSCR) for zero entries corresponding to FO's which can be accomplished during the current mission. For each entry that is found, the following information is printed: proposal number, FO number, mission category, and total possible FO weight. After the entire array has been searched, a summary of the above information is printed by mission category including category number, total possible score of the FO's in that category for which no weight has been accomplished, and total number of FO's in that category for which no weight has been accomplished.

Should a console interrupt occur during the printing of individual FO's, that printing will be terminated, but the summary will be given.

Subroutine PASBAD

Purpose: Determine if the usable portion of a pass over, or in view of a site is acceptable in terms of the length of the usable pass compared to the size of the site. PASBAD is called by FOSCAN to determine whether bad lighting unduly restricts the length of the pass. The other calling routines use PASBAD to determine whether a selected Z-local vertical arc unduly restricts the usable pass.

Calling Sequence: Call PASBAD (TSTART, TEND, START, END, IGOOD) where

TSTART is the value of theta at which the orbit starts a pass over, or in view of, the site (degrees) (input)

TEND is the value of theta at wich the orbit concludes a pass over, or in view of, the site (degrees) (input)

START is the value of theta at the start of a specified arc of the rev (degrees) (input)

END is the value of theta at the end of a specified arc of the rev (degrees) (input)

IGOOD = 0, specified arc does not cover enough of the site for the pass to be acceptable.

= 1, specified arc does cover enough of the site for the pass to be acceptable.

Common Sections Used: CNTROL

Calls: None

Called By: FOSCAN, THEMAX, WGTADD, PSCRMC

Description: If either (END-START) $\geq$ 1/2 (TEND-TSTART) or (TEND-TSTART) $\geq$ ARCMIN, IGOOD =1. Otherwise IGOOD = 0.

Subroutine PPASS

Purpose: Print a summary of the pass data accepted so far.

Calling Sequence: Call PPASS

Common Sections Used: SCOREC, CNSTNT

Calls: TIMDAT

Called By: MAIN

Description: For each accepted pass record contained in the PASS array, PPASS prints the rev number, the longitude of the ascending equatorial crossing, the month, day and hour at the start of the pass, theta at the start and end of the pass, and the sun elevation angle at the start and end of the pass.

Subroutine PPROP

Purpose: Print constrained, unconstrained, and accumulated weights for a given proposal, IPROP, as of the current rev.

Calling Sequence: Call PPROP (IHDG, IFOPRN, IPROP)

IHDG = 1, print rev number and column headings

= 0, do not print rev number and column headings (input)

IFOPRN = 1, print constrained, unconstrained and accumulated scores for each FO of the proposal. (input)

= 0, print scores for complete proposal only.

IPROP is the number of the proposal to be analyzed for constrained, unconstrained, and accumulated weights. (input)

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT, PDATA

Calls: None

Called By: MAIN

Description: PPROP scans the EREP array for all FO's of the specified proposal. As each is found, the current SCORE array is searched for a corresponding entry. If one is located, the constrained and unconstrained weights for that FO are added to the array containing the summary information for the complete proposal. If the print flag, IFOPRN, is set to 1, the individual FO scores are also printed.

When the entire EREP array has been scanned for the FO's of the specified proposal, the summary scores for the proposal are printed.

Subroutine PSCRMC

Purpose: Print a summary by mission category of the FO's in the current SCORE array which can be worked on by a pass from THSTRT to THEND.

Calling Sequence: Call PSCRMC

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT, PDATA

Calls: PASBAD

Called By: MAIN

Description: PSCRMC first determines if the output device, designated by the value of the variable IDEVIC, is 'PRNTR' which indicates a high speed printer. If it is, the SCORE matrix is scanned, and, for each entry, the following information is printed: proposal number, FO number, the values of theta at the beginning and end of the ground track pass over, or in view of, the site, the total weight of the FO, the unconstrained weight accomplished on this rev, the constrained weight accomplished on this rev, the mission category of the FO, and the accumulated weight already accomplished on the FO.

Next, regardless of the value of IDEVIC, the SCORE matrix is searched for FO's worked on during a pass from THSTRT to THEND. For each entry in SCORE, subroutine PASBAD is called to determine if the pass is acceptable. If it is acceptable, its accomplished weights are added to a summary table. After the entire SCORE array has been scanned, the summary table is printed, giving the following information for each mission category: total constrained weights accomplished, total unconstrained weights accomplished, total accumulated weight accomplished so far, total weight possible of FO's worked on, and total number of FO's worked on.

Subroutine PTSCRM

Purpose: For the current mission, print the total number of FO's worked on and the accumulated weights accomplished.

Calling Sequence: Call PTSCRM

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT

Calls: None

Called By: MAIN

Description: If the output device indicated by the value of IDEVIC is 'PRNTR', PTSCRM first prints, for each FO applicable to the current mission, the proposal number, the FO number, the total weight accomplished, and the total possible weight.

Regardless of the value of IDEVIC, the following summary is printed for each category of the current mission: the category number, the total accumulated weight for the category, the total weight possible for the category, and the total number of FO's worked on in the category.

Subroutine PTSCRP

Purpose: Print a summary of the accumulated FO weights accomplished during all three missions.

Calling Sequence: Call PTSCRP

Common Sections Used: CNTROL, EREPC, SCOREC, CNSTNT

Calls: None

Called By: MAIN

Description: PTSCRP scans through the entire EREP matrix accumulating weights for each mission category in a summary table. At the conclusion of the scan, it prints for each category the category number, the total weight accomplished, the total possible weight of all of the FO's in the category, and the total number of FO's worked on.

Subroutine SLOPE

Purpose: Given the coordinates of two points in omega-theta space, form the equation of the straight line between them and determine the value of theta at the point on the line where omega is equal to a given value.

Calling Sequence: Call SLOPE (X1, Y1, XZ, YZ, OM, THET)

X1, Y1
X2, Y2 The omega-theta coordinates of the two points defining the line. (input)

OM The longitude of the ascending equatorial crossing of the groundtrack. (input)

THET The value of theta at the intersection of the groundtrack and the straight line. (output)

Common Sections Used: None

Calls: None

Called By: FLYOVR

Description: The slope and the ordinate intercept are determined for the line connecting the points (X1, Y1) and (X2, Y2). The value of THET is then calculated to be the result of adding the ordinate intercept to the product of OM times the slope.

Subroutine SUNEL

Purpose: Solve for the sun elevation angle at a given value of theta.

Calling Sequence: Call SUNEL (BETA, TNOON, THETA, EL, NOSUCH)

where BETA is the angle between the solar vector and the orbit plane (degrees). (input)

TNOON is the value of theta at orbit noon. (input)

THETA is the value of theta at which the sun elevation is requested. (input)

EL is the calculated sun elevation angle at THETA. (output)

NOSUCH = 0, EL has been calculated
= 1, EL is set to zero; no elevation can be determined for the given value of THETA. (output)

Common Sections Used: None

Calls: COS, ACOS, TAN, SIN

Subroutine THEMAX

Purpose: Determine the best location for a z local vertical pass within a revolution in terms of maximizing the FO weight accomplished.

Calling Sequence: Call THEMAX (ARCPAX, MCATEG, THMAX, THMIN, WGTMXC, WGTMXU)

where ARCPAS is the length of pass. (degrees) (input)

MCATEG is the mission cateogry to be maximized. (For example, if MCATEG = 234, all of the FO's in SCORE are used to maximize the pass weight. If MCATEG = 23, the mission FO's in categories 2, 3, and 23 are used to maximize the pass weight, etc.) (input)

THMAX Upper theta boundary of the rev to be scanned for the best pass. (degrees) (input)

THMIN Lower theta boundary of rev to be scanned for the best pass. If THMAX = THMIN = 0, the theta limits are set to THMADE and THMIDE respectively. (degrees) (input)

WGTMXC is the total constrained weight accomplished during the maximum weight pass. (output)

WGTMXU is the total unconstrained weight accomplished during the maximum weight pass. (output)

Common Sections Used: CNTROL, SCOREC, PDATA, PRNT.

Calls: PASBAD

Called By: MAIN, MASCAN

Description: The SCORE array is searched for entries corresponding to the mission category to be maximized. When an entry is found, the number of that entry in SCORE is placed in an array named THETA. After the entire SCORE array has been searched, the pointers in THETA are sorted to point to records in SCORE in ascending order of theta at the start of the pass over a site.

THEMAX then evaluates the constrained and unconstrained weight accomplished assuming the Z local vertical pass starts at the lowest value of theta in THETA and ends at that value plus ARCPAS degrees. This process is repeated for each value of theta in THETA.

Calls to PASBAD identify the entries in SCORE acceptable for the arc. The weights contained in the acceptable entries are added to the array TMAX which contains the constrained weight, unconstrained weight and theta start value for each arc evaluated.

After all possible arcs have been evaluated, the TMAX array is scanned for the entry with the maximum constrained weight. The value of THSTRT in common section PDATA is set to the theta start value of the entry. THEND is set to THSTRT plus ARCPAS degrees. The constrained and unconstrained weights of the entry are returned to the calling routine as WGTMXC and WGTMCU respectively.

Subroutine TIMDAT

Purpose: Convert time expressed as time elapsed from midnight on December 31 to month, days and hours.

Calling Sequence: Call TIMDAT (TIME, MO, IDAY, HR)

where	TIME	is days elapsed from midnight on December 31. (input)
	MO	is integer month. (output)
	IDAY	is integer day (output
	HR	is floating point hours (output)

Common Sections Used: None

Calls: None

Called By: FOSCAN, PPASS

Description: MO is given a value from 1 to 12 depending upon the month in which TIME falls. IDAY is given the value of the integer number of days in the month corresponding to TIME. HR is given the value of the fractional part of TIME multiplied by 24.

Subroutine WGTADD

Purpose: Simulate the effect of having performed an EREP pass.

Calling Sequence: Call WGTADD

Common Sections Used: EREPC, SCOREC, PDATA

Calls: PASBAD, SUNEL

Called By: MAIN

Description: WGTADD scans the SCORE array to find all FO's which can be worked on during a pass defined from THSTRT to THEND. At each iteration it calls PASBAD to determine if the pass is acceptable. If it is, it adds to the TOTSCR entry for that FO, the constrained weight held in the SCORE array.

If the acceptable FO has a requirement for a pass or passes spaced randomly, the program then loops to consider the next FO since nothing else needs to be recorded. The same thing happens if the FO requires total coverage of the site since no logic is currently implemented to handle total coverage. Finally, if repeating coverage of the site is required by the FO, appropriate entries must be made in the PASDAT array. This includes incrementing the counter contained in PASDAT for the FO representing the number of passes scheduled over the site. Also, the date of the current pass is added to the list of dates on which the passes previously accepted for the FO occurred.

After the entire SCORE array has been scanned, the PASS matrix is updated to reflect the current pass being accepted. IPASS, the counter indicating the number of passes which have been accepted, is incremented by one. The corresponding record in PASS is filled with the following data: the number of the rev on which the pass occurs, the longitude of the ascending equatorial crossing, the date at which the rev begins, theta at the start of the pass, theta at the end of the pass, and the sun elevation angles at the start and end of the pass.

KNOWN ANOMALIES AND INCONSISTENCIES

The writers currently know of only four program anomalies though it should be stressed that the available time has precluded thorough program testing.

The most important anomaly involves the times stored in the PASS array which are supposed to be the time of the start of the ZLV pass. The times currently stored (and printed under the PASSUM print option) are actually the times of the ascending equatorial crossing. To correct this error the statement

```
PASS(3,IPASS) = TRAJ(4,LINE)
```
in subroutine WGTADD should be changed to
```
PASS(3,IPASS) = TRAJ(4,LINE)+THSTRT*ORBPER/360.)/1440.
```
The orbit period, ORBPER, should either be defined by the statement
```
ORBPER = 93.283019
```
or by adding the statement
```
INCLUDE BLK4, LIST
```
if ORBPER is defined in subroutine COMMON as discussed below.

The second anomaly also involves subroutine WGTADD. The constrained weight accomplished on a pass is supposed to be multiplied by PERWT before being added to the TOTSCR array but it is not. The following statement should be added before any executable code

```
INCLUDE BLK1, LIST
```

The statement:

```
TOTSCR(NPT) = TOTSCR(NPT) + SCORE(S,I)
```
should be changed to
```
TOTSCR(NPT) = TOTSCR(NPT) + SCORE(S,I)*PERWT.
```

Under the REVSUM option, the heading format does not permit a four significant digit revolution number. The statement:

```
3000 FORMAT(//T2,'SUMMARY OF REV',I3,',etc.
```
should be changed to
```
3000 FORMAT(//T2,'SUMMARY OF REV',I4,',etc.
```

The sun elevation angle data printed under the print suboption PASSUM is improperly formatted if a negative angle is involved. The second line of the 2000 FORMAT statement should be changed from

```
. F6.2,T44,F6.2,T52,F5.2,T60,F5.2)
```
to:
```
. F6.2,T44,F6.2,T52,F6.2,T60,F6.2)
```

There is a potential inconsistency in the program in that certain orbit sensitive parameters are multiply defined in the various subroutines. Since the program could be used for non-nominal Skylab orbits and for altogether different orbits as well (e.g. ERTS), these parameters should be singly defined and carried in the labeled common section CNSTNT. In addition, certain fixed constants should be added to CNSTNT and the definitions deleted from the subroutines. The following steps will accomplish this goal.

The PDP element EREP.COM should be edited to add the necessary constants to the common section CNSTNT. The following line should be added to the BLK4 PROC entry:

```
. PI,EARRAT,PRERAT,DTOR,RE,RORB,XINCL
```

(The description in the section labeled "variables in common Section CNSTNT " implies this has already been done but it hasn't.)

Several statements should be added to subroutine COMMON. It is suggested that they be added following the statement PERWT=1.0. If added at this point, prior to the display of the control variables, orbit sensitive parameters could easily be added to the display if desired. These variables could also be added to NAMELIST/CONTRL/so they could be changed at program initiation (and to NAMELIST/CONTRL/ in the MAIN routine so they could be changed under the CHANGE option). The new statements are:

```
PI=3.14159265
EARRAT=15.04104953    @EARTH ROTATION RATE - DEG/HR
PRERAT=-.21174988     @INERTIAL PRECESSION RATE - DEG/HR
ORBPER=93.283019      @ORBIT NODAL PERIOD - MINUTES
DTOR=.01745329        @DEGREES TO RADIANS
```

```
RTOD=1./DTOR              @RADIANS TO DEGREES
RE=20925738.2             @EARTH RADIUS - FEET
RORB=22353625.3           @SEMI-MAJOR AXIS - FEET
XINCL=50.                 @ORBIT INCLINATION - DEGREES
```

The following changes should be made to the indicated subroutines.

Subroutine ELTHET

Replace the statement:

```
RTOD=180./3.1415927
```

with:

```
INCLUDE BLK4, LIST
```

Subroutine FLYOVR

Add the statement

```
INCLUDE BLK4, LIST
```

Replace the statements beginning EARTH=, DTOR=, and RINCL=; with

```
EARTH=(EARRAT-PRERAT)/(360./(ORBPER/60.))
RINCL=XINCL*DTOR
```

Subroutine LATLON

Remove the two line DATA statements and replace with:

```
INCLUDE BLK4,LIST
SIN50 = SIN(XINCL*DTOR)
COS50 = cos(XINCL*DTOR)
DOMEGA = -(EARRAT-PRERAT)*ORBPER/21600.
```

(DOMEGA is the change in the longitude of the ascending equatorial crossing expressed in degrees (or radians) change per degree (radian) of spacecraft travel.)

Subroutine LONRAN

Remove the two-line data statement and replace with:

```
INCLUDE BLK4,LIST
SINI = SIN(XINCL*DTOR)
COSI = COS(XINCL*DTOR)
```

```
RATIO = RORB/RE
E = (EARRAT-PRERAT)*ORBPER/21600.
```

(E is the negative of the change in longitude of the ascending equatorial crossing.)

Subroutine SUNEL

Replace:

```
RTOD = 180.3.1415927
```

with:

```
INCLUDE BLK4,LIST
```

Subroutine WGTADD

Add the statement

```
INCLUDE BLK4 LIST
```

The orbit parameters are also defined in the auxiliary routine on file EREP.NEWDAT and should be changed if those in the main program are changed. Failing to change the parameters in NEWDAT will not affect the main program since the site corner points are stored in terms of their latitude/longitude coordinates. However, the printout from NEWDAT listing the site corner points in omega/theta coordinates would be meaningless. The statements involved are

```
OMEGAO = 360./1.55471699 - the mean motion of the spacecraft
                           in degrees per hour

REGRES = .21174988 - the negative of the inertial precession
                     of the orbit line of nodes - degrees
                     per hour.

XINCL = 50.0 - the orbit inclination in degrees.
```

When building the SL2DAT, SL3DAT, and SL4DAT trajectory files, the auxiliary routine on file EREP.MISION automatically discards those revolutions for which the absolute value of beta is greater than 50 degrees. If a different limit is desired, MISION must be changed. The statement which discards the revolution reads:

```
IF (ABS(BETSUN).LT.50) GO TO 60
```

If the limit is changed, the only main program change would be the second and third lines of the 600 FORMAT statement in subroutine COMMON. They now read:

```
. T2,'FIRST REV WITH ABS BETA GREATER THAN 50:  ',I4,/
. T2,'LAST REV WITH ABS BETA GREATER THAN 50:  ',I4)
```

The comment cards describing the purpose of the SUNEL and ELTHET subroutines may be confusing. The code was changed slightly (to adapt them to this program) since the comment cards were written and they were not updated. The purpose of SUNEL should read:

```
PURPOSE:  GIVEN BETA., AND THE VALUE OF THETA AT ORBIT NOON,
          SOLVE FOR THE SUN ELEVATION ANGLE AT THE GIVEN
          VALUE OF THETA
```

The description of ELTHET should read

```
PURPOSE:  GIVEN BETA, AND THE VALUE OF THETA AT ORBIT NOON,
          SOLVE FOR THE VALUES OF THETA FOR A GIVEN SUN
          ELEVATION ANGLE
```

The subroutine VROTAT is included on the program tape but is no longer required. It was used with a previous version of LONRAN.

D. A. Corey
D. A. COREY

D. N. Sakolosky
D. N. SAKOLOSKY

E. W. Radany
E. W. RADANY

DAC
1025-DNS-dvw
EWR

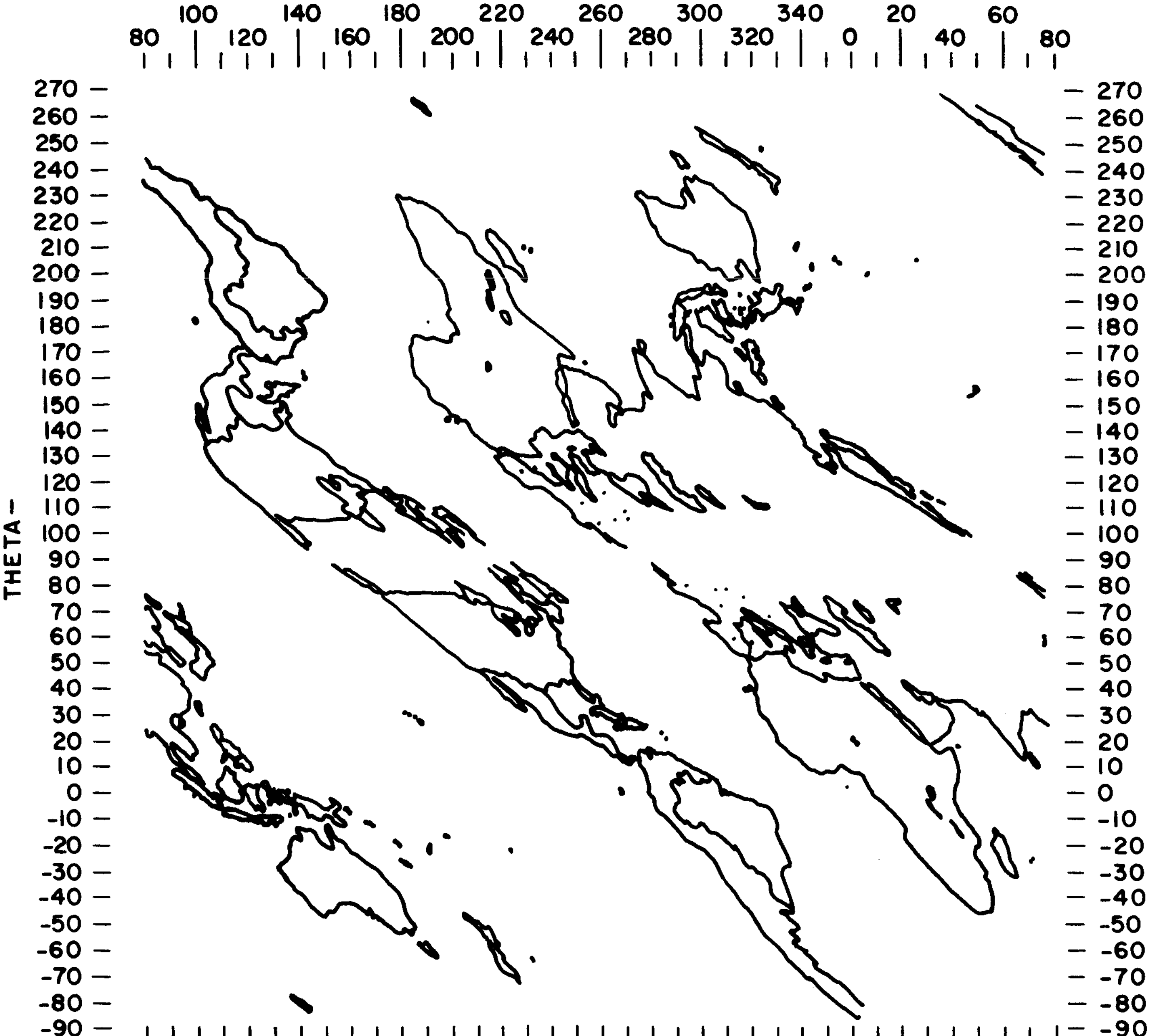

FIGURE I THE WORLD IN OMEGA THETA COORDINATES
50° INCLINATION, 235 NM CIRCULAR ORBIT

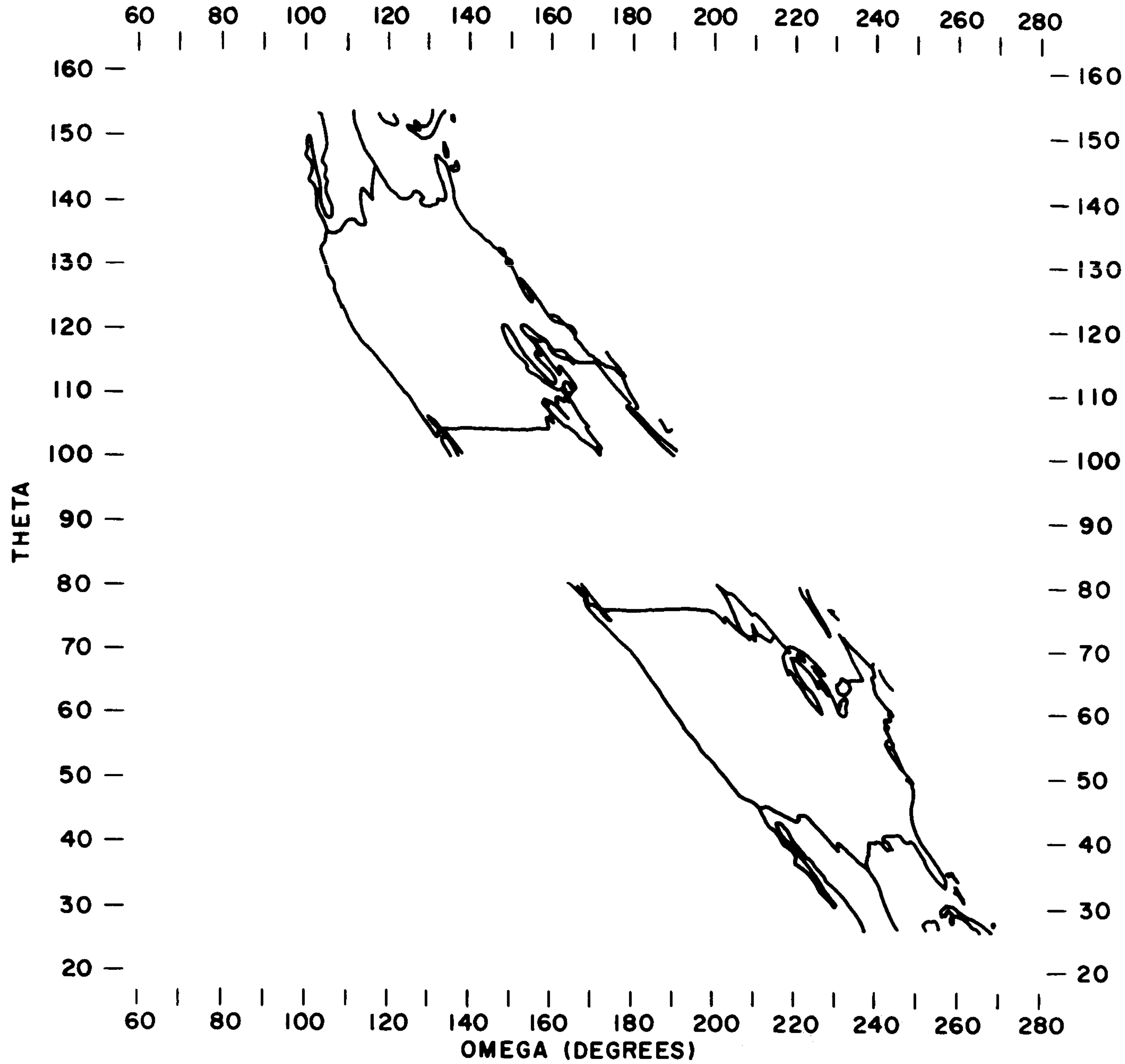

FIGURE 2 THE CONTINENTAL UNITED STATES IN OMEGA THETA COORDINATES 50° INCLINATION, 235 NM CIRCULAR ORBIT

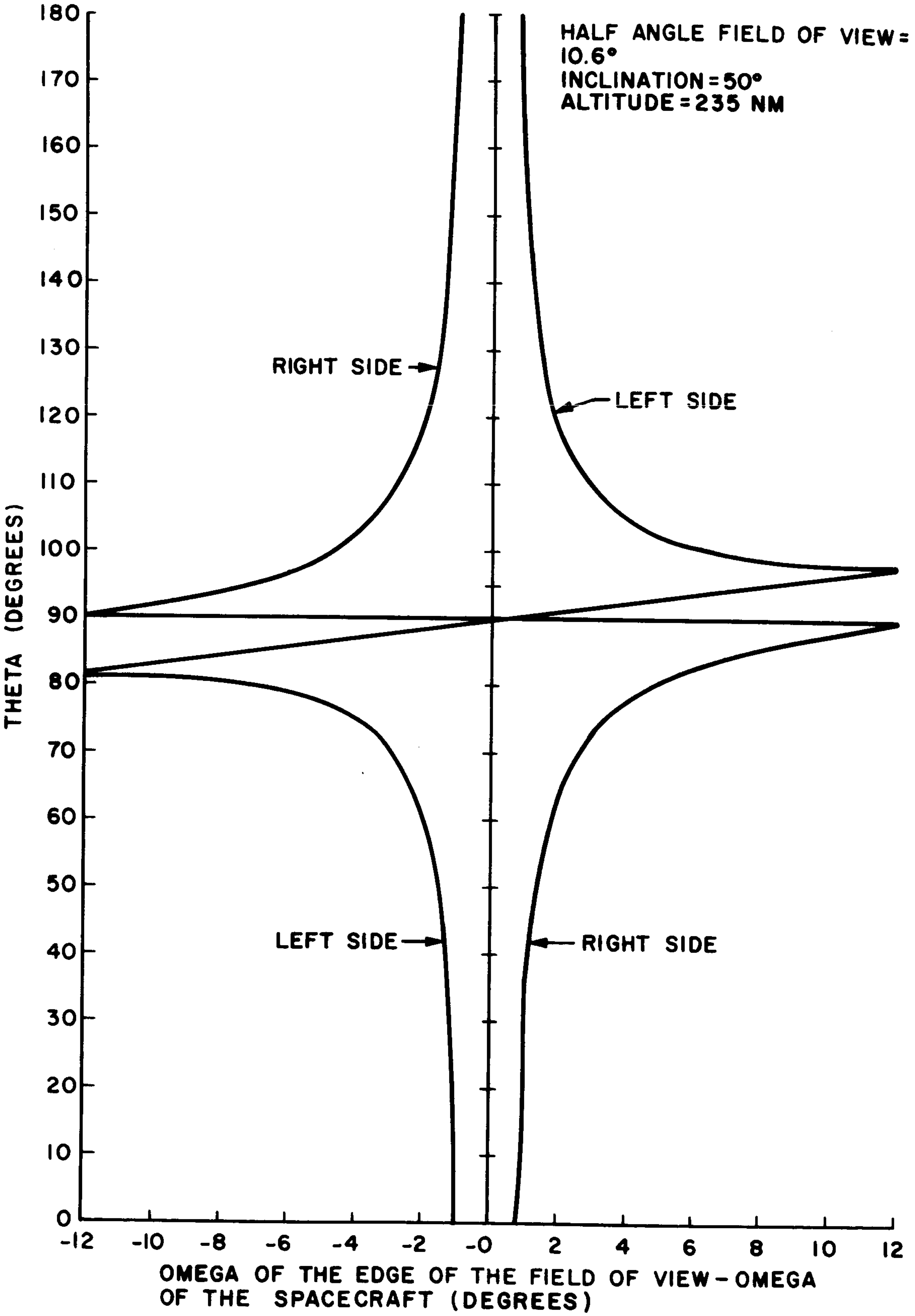

FIGURE 3 THE DIFERENCE BETWEEN THE VALUE OF OMEGA AT THE EDGE OF THE FIELD OF VIEW AND THE VALUE OF OMEGA OF THE SPACECRAFT V.S. THETA FOR THE S190 CAMERAS

APPENDIX A

SUBROUTINE LISTINGS

```
E.ALLOON                                                        08 MAR 72  01:44:32  CYCLE:  0
      SUBROUTINE ALLOCN
C
C   PURPOSE:    LIST THE F.O.'S WHICH HAVE BEEN COMPLETELY
C               ACCOMPLISHED BY ACCEPTED PASSES.  AN ATTENTION
C               BUTTON WILL STOF PRINTING INDIVIDUAL
C               F.O.'S; SUMMARY STILL OBTAINED
C
      INCLUDE BLK1,LIST
      INCLUDE BLK2,LIST
      INCLUDE BLK3,LIST
      INCLUDE BLK4,LIST
C
      DIMENSION PRINT(5,3), IPR(5)
      EQUIVALENCE (PRINT(1,3),IPR(1))
C
      DO 10 J = 1,5
      DO 10 K = 1,3
   10 PRINT(J,K) = 0
      IPFLG = 0
C
      CALL CINOTE(IPFLG)
      WRITE(6,2000)
      DO 100 I = 1,4
      I1 = IEREP(I,1)
      I2 = IEREP(I,2)
      DO 100 J = I1,I2
      IF(TOTSCR(J) .LT. EREP(15,J)) GO TO 100
      PRINT(I,1) = PRINT(I,1) + EREP(15,J)
      PRINT(I,2) = PRINT(I,2) + TOTSCR(J)
      IPR(I) = IPR(I) + 1
      IF(IPFLG .NE. 0) GO TO 100
      NPRO = EREP(1,J)
      NFO = EREP(2,J)
      WRITE(6,2010) NPRO,NFO,MCAT(I),EREP(15,J),TOTSCR(J)
  100 CONTINUE
      WRITE(6,2050)
      DO 300 J = 1,4
      DO 200 K = 1,2
  200 PRINT(5,K) = PRINT(5,K) + PRINT(J,K)
  300 IPR(5) = IPR(5) + IPR(J)
      WRITE(6,3000) ((MCAT(J),(PRINT(J,K),K = 1,2),IPR(J)),J = 1,4),
     . (PRINT(5,K),K = 1,2), IPR(5)
      RETURN
 2000 FORMAT(//T2,'COMPLETED F.O.S'//
     . T2,'PROP',T8,'FO',T13,'MCAT',T19,'POS SCR',
     . T29,'ACCUM SCR')
 2010 FORMAT(T2,I4,T8,I2,T13,I3,T20,F6.2,T31,F6.2)
 2050 FORMAT(//T2,'SUMMARY'/T2,'MISSION CAT',
     . T15,'TOT POS SCR',T29,'TOT ACCUM SCR',
     . T45,'NUM POS SCHLD')
 3000 FORMAT(4(T6,I3,T17,F7.1,T32,F7.1,T50,I3/),
     . T6,'TOTAL',T17,F7.1,132,F7.1,T49,I4)
      END
```

```
E.CINOTE                                                        29 FEB 72  10:52:23  CYCLE:  0
    1..
    2.. TITLE:       CINOTE
    3..
    4.. AUTHOR:      C. MEE
    5..
    6.. DATE:        4/13/70
    7..
    8.. LANGUAGE:    1108 ASSEMBLER
    9..
   10.. DESCRIPTION: CINOTE AND ITS COMPANION ROUTINE, CIJUMP, PERMIT FORTRAN
   11..              USER PROGRAMS TO MAKE USE OF THE CONSOLE INTERRUPT
   12..              (PRESSING THE ATTN KEY ON A REMOTE TERMINAL).
   13..
   14..              CINOTE ALLOWS A FORTRAN USER TO ESTABLISH A VARIABLE IN
   15..              HIS PROGRAM AS A FLAG TO BE SET TO 1 WHENEVER A CONSOLE
   16..              INTERRUPT OCCURS; EXECUTION THEN CONTINUES AS IF THE
   17..              INTERRUPT HAD NOT OCCURRED.  THE USER IS RESPONSIBLE FOR
   18..              TESTING THE FLAG AT APPROPRIATE POINTS WITHIN HIS PROGRAM
   19..              AND FOR CLEARING IT AFTER AN INTERRUPT HAS BEEN DETECTED.
   20..              CINOTE IS CALLED VIA A FORTRAN STATEMENT OF THE FORM
   21..
   22..                           CALL CINOTE(FLAG)
   23..
   24..              WHERE FLAG IS THE VARIABLE TO BE USED AS A FLAG TO RECORD
   25..              INTERRUPTS.
   26..
   27..              CIJUMP PERMITS THE FORTRAN USER TO ESTABLISH A NUMBERED
   28..              STATEMENT WITHIN HIS PROGRAM AS THE LOCATION TO WHICH
   29..              CONTROL IS TO BE TRANSFERRED WHEN A CONSOLE INTERRUPT
   30..              OCCURS.  (THE WRITE-UP FOR SUBROUTINE CIJUMP GIVES
   31..              FURTHER DETAILS.)
   32..
   33..              BOTH CIJUMP AND CINOTE MAY BE CALLED ANY NUMBER OF TIMES.
   34..              EACH CALL OVERRIDES THE EFFECT OF THE PREVIOUS CALL TO
   35..              EITHER ROUTINE.
   36..
   37..              SINCE CINOTE ALLOWS THE USER PROGRAM TO RETAIN CONTROL
   38..              ON A CONSOLE INTERRUPT, PROGRAMS WHICH USE THIS ROUTINE
   39..              CANNOT BE TERMINATED SIMPLY BY PRESSING THE ATTN BUTTON.
   40..              A TASK WHICH GETS OUT OF CONTROL, FOR EXAMPLE BY GETTING
   41..              INTO AN UNWANTED LOOP, CAN BE TERMINATED BY ASKING THE
   42..              COMPUTER OPERATOR TO GIVE THE JOB AN X KEY-IN.
   43..
   44..              CINOTE IS COMPATIBLE WITH AND MAKES USE OF THE
   45..              CONTINGENCY ROUTINE IN THE BELLCOMM FORTRAN I/O PACKAGE,
   46..              BCMIO.
   47..
   48.        AXR$ .
   49..
   50.$(1),CINOTE* .
   51.        L         A0,0,X11 .          GET ADDRESS OF USER'S FLAG
   52.        S         A0,FLAGAD .         AND SAVE FOR LATER USE.
   53.        L,U       A0,INTAD .          SET UP INTAD TO RECEIVE CONTROL
   54.        S,H2      A0,NCONS$ .         IF ATTN KEY IS HIT.
   55.        J         2,X11 .             RETURN TO CALLER.
   56..
```

```
INTAD   L,U      A0,1 .                    SET USER'S FLAG = 1.
        S        A0,*FLAGAD .
        L,H2     A0,NERR$ .                COMPUTE LOCATION
        S        A0,NEXTI .                AND SAVE FOR LATER USE.
        L        A0,NCA0$ .                RESTORE A0.
        L        X11,NCX11$ .              RESTORE X11.
        E        CEND$ .                   ENABLE SUBSEQUENT CONTINGENCIES.
        J        *NEXTI .                  GO TO USER'S NEXT INSTRUCTION.
.
S(0),FLAGAD +    0 .                       ADDRESS OF USER'S FLAG.
NEXTI   +        0 .                       ADDRESS OF USER'S NEXT INSTRUCTION.
.
        END .
```

```
E.COM                                                               07 MAR 72  22:32:40  CYCLE:  0
   1.BLK1    PROC
   2.C
   3.        COMMON/CNTROL/APASDE,MCATDE,THMIDE,THMADE,IDEVIC,MISION,
   4.      .         IERSAV,IEROLD,PERWT,ARCMIN
   5.C
   6. END
   7.BLK2    PROC
   8.C
   9.        COMMON/EREPC/EREP(15,1000),INDEX(2,600),SITE(2,5000),
  10.      .         TRAJ(5,700),ITOTRJ, IERPNT(7,2), NBETA1, NBETA2,
  11.      .         LINE
  12.C
  13. END
  14.BLK3    PROC
  15.C
  16.        COMMON/SCOREC/PASS(7,200),TOTSCR(1000),PASDAT(12,200),
  17.      .         OMECOV(13,200),NSCORE,SCORE(6,100), IPASS, IDAY
  18.C
  19. END
  20.BLK4    PROC
  21.C
  22.        COMMON/CNSTNT/RTOD,NUMPRO,MCAT(4),PRLIST(200),ORBPER,IEREP(4,2),
  23.      . HAS190,HAS191,HAS192,HAS193,HAS194
  24.C
  25. END
  26.BLK5    PROC
  27.C
  28.        COMMON/PDATA/IREV,THSTRT,THEND
  29.C
  30. END
```

```
E.COMMON                                                         08 MAR 72  01:45:40  CYCLE:  0
          SUBROUTINE COMMON(ICHFLG)
          INCLUDE BLK1,LIST
          INCLUDE BLK2,LIST
          INCLUDE BLK3,LIST
          INCLUDE BLK4,LIST
          NAMELIST/CONTRL/APASDE,MCATDE,THMIDE,THMADE,IDEVIC,
     .              MISION,IERSAV,IEROLD,PERWT,ARCMIN
C
          IF(ICHFLG .NE. 0) GO TO 100
          APASDE = 60.0
          MCATDE = 234
          THMIDE = -90.0
          THMADE = 270.0
          IDEVIC = 'TERMNL'
          MISION = 2
          IERSAV = 0
          IEROLD = 0
          ARCMIN = 1.0
          PERWT = 1.0
          WRITE(6,5000) APASDE,MCATDE,MISION,THMIDE,THMADE,PERWT,
     .              IERSAV,IEROLD,IDEVIC,ARCMIN
          READ(5,5010) IANS
          IF(IANS .NE. 'NO') GO TO 10
          WRITE(6,5020)
          READ(5,CONTRL)
   10     READ(10) IERPNT,EREP
          READ(11) NENTR, INDEX,SITE
          IF(IEROLD .EQ. 0) GO TO 100
          READ(IEROLD) TOTSCR,PASDAT,OMECOV,PASS, IPASS
  100     READ(MISION) ITOTRJ, NBETA1, NBETA2, TRAJ
          IF(MISION .NE. 2) GO TO 130
          DO 125 J = 1,4
          DO 125 K = 1,2
  125     IEREP(J,K) = IERPNT(J,K)
          MCAT(1) = 2
          MCAT(2) = 23
          MCAT(3) = 24
          MCAT(4) = 234
          GO TO 150
C
  130     IF(MISION .NE. 3) GO TO 140
          IEREP(1,1) = IERPNT(2,1)
          IEREP(1,2) = IERPNT(2,2)
          IEREP(2,1) = IERPNT(4,1)
          IEREP(2,2) = IERPNT(4,2)
          IEREP(3,1) = IERPNT(5,1)
          IEREP(3,2) = IERPNT(5,2)
          IEREP(4,1) = IERPNT(6,1)
          IEREP(4,2) = IERPNT(6,2)
          MCAT(1) = 23
          MCAT(2) = 234
          MCAT(3) = 3
          MCAT(4) = 34
          GO TO 150
C
  140     IEREP(1,1) = IERPNT(3,1)
```

```
      IEREP(1,2) = IERPNT(3,2)
      IEREP(2,1) = IERPNT(4,1)
      IEREP(2,2) = IERPNT(4,2)
      IEREP(3,1) = IERPNT(6,1)
      IEREP(3,2) = IERPNT(6,2)
      IEREP(4,1) = IERPNT(7,1)
      IEREP(4,2) = IERPNT(7,2)
      MCAT(1) = 24
      MCAT(2) = 234
      MCAT(3) = 34
      MCAT(4) = 4
C
  150 ICHFLG = 0
      HAS190 = 10.6
      HAS191 = 20.
      HAS192 = 5.
      HAS193 = 12.4
      HAS194 = 7.5
      WRITE(6,6000) TRAJ(1,1), TRAJ(1,ITOTRJ), NBETA1, NBETA2
      I = IEREP(1,1)
      NUMPRO = 1
      PRLIST(1) = EREP(1,I)
      DO 400 I = 1,4
      ISTART = IEREP(I,1)
      IEND = IEREP(I,2)
      DO 400 J = ISTART, IEND
      IF(EREP(1,J) .EQ. PRLIST(NUMPRO)) GO TO 400
      IF(NUMPRO .GE. 200) GO TO 500
      NUMPRO = NUMPRO + 1
      PRLIST(NUMPRO) = EREP(1,J)
  400 CONTINUE
      RETURN
  500 WRITE(6,6050)
      RETURN
C
C   FORMATS
C
 5000 FORMAT(/T2,'CURRENT VALUES OF CONTROL VARIABLES ARE:'/
     . T2,'APASDE = ',F9.4,T22,'MCATDE = ',I3,
     . T42,'MISION = ',I3/
     . T2,'THMIDE = ',F9.4,T22,'THMADE = ',F9.4,
     . T42,'PERMT = ',F7.3/
     . T2,'IERSAV = ',I3,T22,'IEROLD = ',I3,T42,'IDEVIC = ',A6/
     . T2,'ARCMIN = ', F6.3,5X,'OK?')
 5010 FORMAT(A6)
 5020 FORMAT(/T2,'ENTER CHANGED PARAMETERS, NAMELIST CONTRL')
 6000 FORMAT(//T2,'FIRST REV = ',F5.0,5X,'LAST REV = ',F5.0/
     . T2,'FIRST REV WITH ABS BETA GREATER THAN 50:  ',I4/
     .  T2,'LAST REV WITH ABS BETA GREATER THAN 50:  ',I4)
 6050 FORMAT(/T2,'*****ERROR   NUMBER OF PROPOSALS ',
     . 'EXCEEDS DIMENSION OF PRLIST ARRAY.')
      END
```

```
E.ELTHET                                              26 JAN 72  14:00:19  CYCLE:  0
            SUBROUTINE ELTHET(BETA,TNOON,EL,THETUP,THETDN,NOSUCH)
C
C        PURPOSE:  GIVEN BETA, AND THE VALUES OF THETA WHEN THE SUN
C                  ELEVATION ANGLE IS 30 DEG. AND INCREASING AND 30 DEG.
C                  AND DECREASING (OR ANY SIMILAR PAIR), SOLVE FOR THE
C                   VALUES OF THETA FOR A GIVEN SUN ELEVATION ANGLE.
C
C
            RTOD=180./3.1415927
            NOSUCH=0
            X=SIN(EL/RTOD)/COS(BETA/RTOD)
            IF(ABS(X).GT.1)GO TO 30
            X=ACOS(X)*RTOD
            THETUP=TNOON-X
            THETDN=TNOON+X
            IF(THETUP .LT. -90.) THETUP = THETUP + 360.
            IF(THETDN .GT. 270.) THETDN = THETDN - 360.
            RETURN
      30    NOSUCH=1
            THETUP=0.
            THETDN=0.
            RETURN
            END
```

```
                                                                    2U MAR 72  13:27:11  CYCLE:  0
E.ERMAP
    1.        LIB        EREP
    2.        I          MAIN
```

```
                                                                   22 FEB 72  12:32:37  CYCLE:  0
E.FILES
    1.@ASG,AX            SL2DAT
    2.@USE    2,SL2DAT
    3.@ASG,AX                                    SL3DAT
    4.@USE                                       3,SL3DAT
    5.@ASG,AX            SL4DAT
    6.@USE    4,SL4DAT
    7.@ASG,AX            ERDATA
    8.@USE    10,ERDATA
    9.@ASG,AX            LATLON
   10.@USE    11,LATLON
   11.@ASG,T            TFILE.,F
   12.@USE   15,TFILE
```

```
E.FLYOVR                                                16 MAR 72  14:58:18  CYCLE:  0
      SUBROUTINE FLYOVR(OMEGA,ISITE,FOV,ITOTAL,IOVER,TSTRT,TEND,
     *                  OMMIN,OMMAX)
C
C   PURPOSE:   DETERMINE WHETHER A REV FLIES OVER A SITE.  IF SO,
C              COMPUTE THETA LIMITS OF PASS AND, IF TOTAL COVERAGE IS
C              REQUIRED, THE LIMITS OF OMEGA EFFECTIVELY COVERED.
C
      INCLUDE BLK2,LIST
      DIMENSION TSTRT(2), TEND(2), OMMIN(2), OMMAX(2), OMETH(30,4),
     *            TH(30)
C
C     CONVERT SITE CORNER POINTS TO OMEGA AND THETA,
C     ASCENDING AND DESCENDING
C
      NSITE = ISITE
      EARTH = (15.04104953 + .21174988)/(360./1.55471699)
      DTOR = 3.14159265 / 180.
      RINCL = 50.0 * DTOR
      SINRIN = SIN(RINCL)
      PI = 3.14159265
      NPTS = INDEX(2,ISITE)
      I1 = INDEX(1,ISITE)
      I2 = I1 + NPTS - 1
    5 DO 20 I = 1,NPTS
      N = I1 + I - 1
      IF(ABS(SITE(1,N)) .LT. 50.0) GO TO 15
      IF(IOUT .NE. 1) GO TO 25
      IF(SITE(1,N) .LT. 0) SITE(1,N) = -49.99
      IF(SITE(1,N) .GT. 0) SITE(1,N) = 49.00
   15 OMETH(I,1) = ASIN(SIN(SITE(1,N) * DTOR)/SINRIN)
      OMETH(I,2) = PI - OMETH(I,1)
      DO 17 J = 3,4
   17 OMETH(I,J) = SITE(2,N) * DTOR - ATAN2(COS(RINCL) *
     * SIN(OMETH(I,J-2)),COS(OMETH(I,J-2))) + EARTH * OMETH(I,J-2)
      DO 18 J = 1,4
   18 OMETH(I,J) = OMETH(I,J)/DTOR
      DO 19 J = 3,4
      IF(OMETH(I,J) .LT. 0) OMETH(I,J) = OMETH(I,J) + 360.
   19 IF(OMETH(I,J) .GT. 360) OMETH(I,J) = OMETH(I,J) - 360.
   20 CONTINUE
      IOUT = 0
      GO TO 50
C
C     IF ALL POINTS OF SITE ARE OUT OF RANGE, IGNORE SITE.
C     IF SOME POINTS ARE IN RANGE, SET ABS. VAL. OF OUTSIDE
C     LATITUDES = 49.99
C
   25 DO 30 I = I1,I2
   30 IF(ABS(SITE(1,I)) .LT. 50) GO TO 40
      RETURN
   40 IOUT = 1
      GO TO 5
   50 IF(OMETH(1,1) .EQ. OMETH(NPTS,1) .AND.
     * OMETH(1,3) .EQ. OMETH(NPTS,3) .AND. NPTS .NE. 1) GO TO 60
      NPTS = NPTS + 1
      DO 55 J = 1,4
   55  OMETH(NPTS,J) = OMETH(1,J)
C
C      DETERMINE MAX AND MIN OMEGAS FOR ASCENDING AND DESCENDING PASS
C
   60  OMMIN(1) = OMETH(1,3)
       OMMAX(1) = OMETH(1,3)
       DO 65 J = 1,NPTS
       OMMIN(1) = AMIN1(OMMIN(1),OMETH(J,3))
   65  OMMAX(1) = AMAX1(OMMAX(1),OMETH(J,3))
       IF((OMMAX(1) - OMMIN(1)) .LT. 180 .OR. I180 .EQ. 1)
     . GO TO 75
       DO 70 J = 1,NPTS
   70  IF(OMETH(J,3) .LT. 180) OMETH(J,3) = OMETH(J,3) + 360.
       I180 = 1
       GO TO 60
   75  I180 = 0
   80  OMMIN(2) = OMETH(1,4)
       OMMAX(2) = OMETH(1,4)
       DO 85 J = 1,NPTS
       OMMIN(2) = AMIN1(OMMIN(2),OMETH(J,4))
   85  OMMAX(2) = AMAX1(OMMAX(2),OMETH(J,4))
       IF((OMMAX(2) - OMMIN(2)) .LT. 180 .OR. I180 .EQ. 1)
     . GO TO 110
       DO 90 J = 1,NPTS
   90  IF(OMETH(J,4) .LT. 180) OMETH(J,4) = OMETH(J,4) + 360.
       I180 = 1
       GO TO 80
  110  I180 = 0
       DO 500 K = 1,2
       J = 0
C
C      DETERMINE AVERAGE THETA
C
       THMIN = OMETH(1,K)
       THMAX = THMIN
       DO 120 J = 1,NPTS
       THMIN = AMIN1(THMIN,OMETH(J,K))
  120  THMAX = AMAX1(THMAX,OMETH(J,K))
       THAVG = (THMAX + THMIN)/2.
C
C      CALCULATE DELTA OMEGA LEFT AND RIGHT
C
       CALL LONRAN(THAVG,FOV,DOML,DOMR)  @  CHECK ARGUMENTS
       FOVOM = DOMR - DOML
C
C      CHECK FOR POINT SITE
C
       IF(NPTS .NE. 2) GO TO 125
       IF(OMMIN(K) .GE. (OMEGA + DOML) .AND.
     . OMMAX(K) .LE. (OMEGA + DOMR)) GO TO 122
       IF((OMMIN(K) + 360.) .GE. (OMEGA + DOML) .AND.
     . (OMMAX(K) + 360.) .LE. (OMEGA + DOMR)) GO TO 122
       GO TO 500
  122  TSTRT(K) = THMIN
       TEND(K) = THMAX
       IOVER = IOVER + K
       GO TO 500
```

```
C
C      CHECK FOR OVERFLIGHT OF NON-POINT SITE
C
  125  IF(OMEGA .GE. OMMIN(K) .AND. OMEGA .LE. OMMAX(K)) GO TO 130
       IF((OMEGA + 360.) .GE. OMMIN(K) .AND. (OMEGA + 360.) .LE.
     . OMMAX(K)) GO TO 140
       GO TO 300
C
C      REV GOES OVER SITE
C
  130  OM = OMEGA
       GO TO 145
  140  OM = OMEGA + 360
  145  J = 1
       N1 = NPTS - 1
       DO 160 I = 1,N1
       IF(OMETH(I,K+2) .GT. OMETH(I+1,K+2)) GO TO 150
       IF(OM .LE. OMETH(I,K+2) .OR. OM .GT. OMETH(I+1,K+2)) GO TO 160
       GO TO 155
  150  IF(OM.LE. OMETH(I+1,K+2) .OR. OM .GT. OMETH(I,K+2)) GO TO 160
  155  CALL SLOPE(OMETH(I,K+2),OMETH(I,K),OMETH(I+1,K+2),
     . OMETH(I+1,K),OM,TH(J))
       J = J + 1
  160  CONTINUE
       J = J - 1
       TSTRT(K) = TH(1)
       TEND(K) = TH(1)
       DO 170  I = 1,J
       TSTRT(K) = AMIN1(TH(I),TSTRT(K))
  170  TEND(K) = AMAX1(TH(I),TEND(K))
C
C      UPDATE IOVER
C
       IOVER = IOVER + K
       IF(ITOTAL .NE. 1) GO TO 500
       IF(OMMIN(K) .LT. (OM + DOML)) OMMIN(K) = OM + DOML
       IF(OMMAX(K) .GT. (OM + DOMR)) OMMAX(K) = OM + DOMR
       GO TO 500
C
C      GROUND TRACK DOES NOT PASS OVER SITE.  IF SITE IS LARGE,
C      GROUND TRACK MUST PASS OVER TO COUNT.  IF SITE IS SMALL,
C      DETERMINE IF AT LEAST 1/4TH OF SITE IS WITHIN THE FIELD
C      OF VIEW.
C
  300  IF((OMMAX(K) - OMMIN(K)) .GE. FOVOM) GO TO 500
       QOMR = OMMAX(K) - ((OMMAX(K) - OMMIN(K))/4.)
       QOML = OMMIN(K) + ((OMMAX(K) - OMMIN(K))/4.)
       OM = OMEGA
       IF(OM .LT. 0) OM = OM + 360
       IF((OM + DOML) .GT. QOMR) GO TO 500  @  OMEGA TOO LARGE
       IF((OM + DOMR) .LT. QOML) GO TO 500  @  OMEGA TOO SMALL
C
C      SMALL SITE IS COVERED
C
  310  IOVER = IOVER + K
       IF(OM .GT. QOMR) GO TO 320
       DOM = OM + DOMR                   @  SITE TO RIGHT
```

```
      IF (DUM .GE. OMMAX(K)) GO TO 371
      GO TO 330
320   DUM = OM + DOML                @ SITE TO LEFT
      IF (DUM .LE. OMMIN(K)) GO TO 371
330   N1 = NPTS - 1
      J = 0
      DO 350 I = 1,N1
      IF(OMETH(I+1,K+2) .LT. OMETH(I,K+2)) GO TO 340
      IF(DUM .LT. OMETH(I,K+2) .OR.
    . DUM .GT. OMETH(I+1,K+2)) GO TO 350
      GO TO 345
340   IF(DUM .LT. OMETH(I+1,K+2) .OR. DUM .GT. OMETH(I,K+2)) GO TO 350
345   J = J+1
      CALL SLOPF(OMETH(I,K+2),OMETH(I,K),OMETH(I+1,K+2),
    . OMETH(I+1,K),DUM,TH(J))
350   CONTINUE
      IF(J .GE. 2) GO TO 360
      WRITE(6,1200) NSITE
      RETURN
360   TSTRT(K) = TH(1)
      TEND(K) = TH(1)
      DO 370 I = 1,J
      TSTRT(K) = AMAX1(TSTRT(K),TH(I))
370   TEND(K) = AMIN1(TEND(K),TH(I))
      GO TO 372
371   TSTRT(K)=THMIN     @ ENTIRE SITE WILL BE IN VIEW
      TEND(K)=THMAX      @ THETA RANGE SET TO COVER THE WHOLE SITE
372   IF(ITOTAL .NE. 1) GO TO 500
      IF(OM .LE. OMMAX(K)) GO TO 380
      IF((OM + DOML) .LT. OMMIN(K)) GO TO 500
      OMMIN(K) = OM + DOML
      GO TO 500
380   IF((OM + DOMR) .GT. OMMAX(K)) GO TO 500
      OMMAX(K) = OM + DOMR
500   CONTINUE
      RETURN
1200  FORMAT(/T2,'PASS ENTERED OR LEFT SITE ',I4,' BUT NOT BOTH',
    . '*****ERROR*****')
      END
```

```
E.FOSCAN                                                        06 MAR 72  09:42:32  CYCLE:  0
         SUBROUTINE FOSCAN

      PURPOSE:    SCAN THE EREP MATRIX TO DETERMINE WHICH F.O.'S CAN
                  BE SATISFIED ON THE GIVEN REVOLUTION.

         DIMENSION IS19(5), TSTRT(2), TEND(2), OMMIN(2), OMMAX(2),
       . THES(2), THEE(2), START(2), END(2)
         INCLUDE BLK1,LIST
         INCLUDE BLK2,LIST
         INCLUDE BLK3,LIST
         INCLUDE BLK4,LIST
         INCLUDE BLK5,LIST


         IPFLG = 0
         CALL CIMOTE(IPFLG)


         IF(NBETA1 .EQ. 0 .AND. NBETA2 .EQ. 0) GO TO 5
         IF(IREV .LT. NBETA1 .AND. IREV .GE. TRAJ(1,1)) GO TO 10
         IF(IREV .GT. NBETA2 .AND. IREV .LE. TRAJ(1,ITOTRJ)) GO TO 20
     2     RITE(6,1000) IREV
         RETURN
     5   IF(IREV .GE. TRAJ(1,1) .AND. IREV .LE. TRAJ(1,ITOTRJ)) GO TO 10
         GO TO 2
    10   LINE = IREV - TRAJ(1,1) + 1
         GO TO 30
    20   LINE = IREV - TRAJ(1,1) - NBETA2 + NBETA1
    30   CALL TIMDAT(TRAJ(4,LINE),MO,IDAY,HR)

         DO 40 I = 1,6
         DO 40 J = 1,100
    40   SCORE(I,J) = 0
         NSCORE = 0
         DO 950 JJ = 1,4
         ISTART = IEREP(JJ,1)
         ISTOP = IEREP(JJ,2)
         DO 900 J = ISTART, ISTOP
         ICOVER = 0

         CHECK IF MONTH DURING WHICH THIS REV OCCURS IS
         INCLUDED IN THE MONTHS DESIRED FOR THIS F.O.

         IDAT = EREP(6,J)
         M1 = IDAT/100
         M2 = MOD(IDAT,100)
         IF(M2 .LT. M1) GO TO 50
         IF(MO .LT. M1 .OR. MO .GT. M2) GO TO 900
         GO TO 60
    50   IF(MO .LT. M1 .AND. MO .GT. M2) GO TO 900
    60   OMEGA = TRAJ(2,LINE)

         DETERMINE IF TOTAL COVERAGE REQUIRED

         ICOV = EREP(7,J)
         IF(MOD(ICOV,10) .NE. 1) GO TO 70
```

```
          ITOTAL = 1
          GO TO 75
     70   ITOTAL = 0
C
C         DETERMINE WHICH SENSORS REQUIRED.  CHOOSE NARROWEST
C         INSTRUMENT FIELD OF VIEW TO CALCULATE IF SITE OVERFLOWN.
C
     75   DO 80 K = 1,5
     80   IS19(K) = 0
          ISENS = EREP(5,J)
          DO 90 K = 1,5
          IF(MOD(ISENS,10) .EQ. 1) IS19(K) = 1
     90   ISENS = ISENS/10
          IF(IS19(3) .NE. 1) GO TO 100
          FOV = HAS192
          GO TO 150
    100   IF(IS19(5) .NE. 1) GO TO 120
          FOV = HAS194
          GO TO 150
    120   IF(IS19(1) .NE. 1) GO TO 130
          FOV = HAS190
          GO TO 150
    130   IF(IS19(4) .NE. 1) GO TO 140
          FOV = HAS193
          GO TO 150
    140   FOV = HAS191
C
C         DETERMINE IF SITE IS OVERFLOWN
C
    150   ISITE = EREP(3,J)
          CALL FLYOVR(OMEGA,ISITE,FOV,ITOTAL,ICOVER,TSTRT,TEND,
        • OMMIN, OMMAX)
          IF(ICOVER .EQ. 0) GO TO 900
          WRITE(15,7114) TSTRT, TEND
   7114   FORMAT(/T2,'TSTRT = ',2(F8.3,2X),'TEND = ',2(F8.3,2X))
C
C         CALCULATE SUN ELEVATION DURING OVERFLIGHT
C
          THNOON = TRAJ(5,LINE)
          THMID = THNOON + 180.
          IF(THMID .GT. 270.) THMID = THMID - 360.
          CALL SUNEL(TRAJ(3,LINE),THNOON,THNOON,RSNMAX,NOSUCH)
          CALL SUNEL(TRAJ(3,LINE),THNOON,THMID,RSNMIN,NOSUCH)
          IF(RSNMIN .GE. EREP(10,J)) GO TO 900  @  SUN TOO HIGH THIS REV
          IF(RSNMAX .LE. EREP(9,J)) GO TO 900  @  SUN TOO LOW THIS REV
          LSUN = 1
          IF(RSNMAX .GT. EREP(10,J)) GO TO 200
          IF(RSNMIN .LT. EREP(9,J)) GO TO 170
          THES(1) = -90.
          THEE(1) = 270.
          GO TO 250
    170   RSNMIN = EREP(9,J)
          CALL ELTHET(TRAJ(3,LINE),THNOON,RSNMIN,THES(1),THEE(1),NOSUCH)
          GO TO 250
    200   RSNMAX = EREP(10,J)
          IF(RSNMIN .LT. EREP(9,J)) GO TO 220
          CALL ELTHET(TRAJ(3,LINE),THNOON,RSNMAX,THEE(1),THES(1),NOSUCH)
        GO TO 250
  220   RSNMIN = FREP(9,J)
        LSUN = 2
        CALL ELTHET(TRAJ(3,LINE),THNOON,RSNMIN,THES(1),THEE(2),NOSUCH)
        CALL ELTHET(TRAJ(3,LINE),THNOON,RSNMAX,THEE(1),THES(2),NOSUCH)
C
C       SET K AND KK:
C                 K = 1  INDICATES ASCENDING LEG
C                 K = 2  INDICATES DESCENDING LEG
C                 KK  KEEPS TRACK OF THE NUMBER OF VALID LIGHTED
C                     PASSES ON THIS REV FOR THIS F.O.
  250   IF(ICOVER .EQ. 1 .OR. ICOVER .EQ. 3) GO TO 260
        K = 2
        GO TO 270
  260   K = 1
  270   KK = 0
C
C       SET L COUNTER.  L KEEPS TRACK OF WHICH LIGHTED PORTION
C       OF THE REV IS BEING CONSIDERED IN CASE LSUN = 2.
C
  300   L = 1
  310   IF(THEE(L) .LT. THES(L)) GO TO 350
        IF(TEND(K) .LT. THES(L) .OR. TSTRT(K) .GT. THEE(L)) GO TO 490
        KK = KK + 1
        IF(TSTRT(K) .LT. THES(L)) GO TO 320
        START(KK) = TSTRT(K)
        GO TO 330
  320   START(KK) = THES(L)
C
C       (CASE A)
C
  330   IF(TEND(K) .GT. THEE(L)) GO TO 340
        END(KK) = TEND(K)
        GO TO 450
  340   END(KK) = THEE(L)
        GO TO 450
C
C       PROPERLY LIGHTED PORTION SPANS 270/-90     (CASE B)
C
  350   IF(TSTRT(K) .LT. THEE(L)) GO TO 400
        IF(TEND(K) .LT. THES(L)) GO TO 490
        KK = KK + 1
        END(KK) = TEND(K)
        IF(TSTRT(K) .LT. THES(L)) GO TO 360
        START(KK) = TSTRT(K)
        GO TO 450
  360   START(KK) = THES(L)
        GO TO 450
C
C       (B2,3,4)
C
  400   KK = KK + 1
        START(KK) = TSTRT(K)
        IF(TEND(K) .GT. THEE(L)) GO TO 410
        END(KK) = TEND(K)                    (B3)
        GO TO 450
  410   END(KK) = THEE(L)
```

```
       IF(TEND(K) .GT. THES(L)) GO TO 420
       GO TO 450
  420  CALL PASBAD(TSTRT(K),TEND(K),START(KK),END(KK),IGOOD)
       IF(IGOOD .EQ. 0) GO TO 430
       KK = KK + 1
C
C      (B4)  DOUBLE PASS
C
  430  START(KK) = THES(L)
       END(KK) = TEND(K)
C
C      (B2)
C
  450  CALL PASBAD(TSTRT(K),TEND(K),START(KK),END(KK),IGOOD)
       IF(IGOOD .NE. 0) GO TO 490
       KK = KK - 1
C
C      NO LIGHTED PASS
C
  490  IF(LSUN .NE. 2 .OR. L .NE. 1) GO TO 500
       L = 2
       GO TO 310
  500  IF(ICOVER .NE. 3 .OR. K .NE. 1) GO TO 510
       K = 2
       GO TO 300
C
C      LOGIC DEALING WITH SPECIAL NOTES BELONGS HERE AT 510
C
  510  CONTINUE
C
C
       IF(KK .GT. 1) GO TO 550
       IF(KK .EQ. 0) GO TO 900
C
C      DETERMINE THE LARGEST ARC (END(K) - START(K))
C      AND PUT THE POINTS IN END(1), START(1)
C
  550  IF(MOD(ICOV,10) .EQ. 0) GO TO 560
       IF(MOD(ICOV,10) .EQ. 1) GO TO 600
       GO TO 560
C
C      (NOTHING DONE YET ON TOTAL COVERAGE)
C
C
C      ONE OR MORE SINGLE PASSES REQUIRED
C
  555  INOREP = 1
  560  NUM = ICOV/1000
       IF(NSCORE .LT. 100) GO TO 565
       WRITE(6,4500)
       RETURN
  565  NSCORE = NSCORE + 1
       SCORE(4,NSCORE) = EREP(15,J)/NUM
       IF(INOREP .EQ. 1) GO TO 580
       IF(SCORE(4,NSCORE) + TOTSCR(J) .GT. EREP(15,J)) GO TO 570
       SCORE(5,NSCORE) = SCORE(4,NSCORE)
       GO TO 800
  570 SCORE(5,NSCORE) = AMAX1(0.,(EREP(15,J) - TOTSCR(J)))
      GO TO 800
  580 SCORE(5,NSCORE) = 0
      INOREP = 0
      GO TO 800
C
C     REPEATING COVERAGE
C
  600 NDAYS = MOD(ICOV/10,100)
      IPCOL = 0
  610 IPCOL = IPCOL + 1
      IF(PASDAT(1,IPCOL) .EQ. J) GO TO 630
      IF(BOOL(PASDAT(1,IPCOL)) .EQ. 0) GO TO 560
      IF(IPCOL .EQ. 200) GO TO 620
      GO TO 610
  620 WRITE(6,1500) J
      RETURN
  630 IDAY = TRAJ(4,LINE)
      ILAST = PASDAT(2,IPCOL)
      DO 640  N = 1,ILAST
      IPDAY = PASDAT(N+2,IPCOL)
  640 IF(IPDAY .EQ. IDAY) GO TO 555
      DO 650 N = 1,ILAST
      DUM = ABS(TRAJ(4,LINE) - PASDAT(N+2,IPCOL))/NDAYS
      NDUM = DUM
      DUM = DUM - NDUM
  650 IF((DUM .GE. 0 .AND. DUM .LE. .1) .OR.
     . (DUM .GE. .9 .AND. DUM .LE. 1.)) GO TO 560
      GO TO 555
  800 SCORE(1,NSCORE) = J
      SCORE(2,NSCORE) = START(I)
      SCORE(3,NSCORE) = END(I)
      SCORE(6,NSCORE) = MCAT(JJ)
  900 CONTINUE
  950 CONTINUE
      RETURN
 1000 FORMAT(//T2,'SPECIFIED REV EITHER DOES NOT OCCUR IN CURRENT ',
     . 'MISSION OR OCCURS WHEN ABS BETA  GREATER THAN 50 DEGREES')
 1500 FORMAT(//T2,'THE F.O. IN ENTRY ',I4,' OF THE EREP ARRAY ',
     . 'IS NOT LISTED IN PASDAT.')
 4500 FORMAT(/T2,'ENTRIES IN SCORE EXCEED DIMENSION OF ARRAY'
     . '*****ERROR')
      END
```

01 FEB 72 11:00:39 CYCLE: 0

E.LATLON

```
      SUBROUTINE LATLON(XLATC,XLON,OMEGA1,OMEGA2)
      DATA SIN50/.766044443/RTOD/57.2957795/PI/3.14159265/
     .COS50/.642787610/DOMEGA/-.065871629/
      RATIO=SIN(XLATC/RTOD)/SIN50
      IF(RATIO.GT.1.)RATIO=1.
      IF(RATIO.LT.-1.)RATIO=-1.
      THETA1=RTOD*ASIN(RATIO)
      OMEGA1=XLON-RTOD*ATAN(COS50*TAN(THETA1/RTOD))
     .-DOMEGA*THETA1
      THETA2=180.-THETA1
      OMEGA2=XLON-RTOD*ATAN(COS50*TAN(THETA2/RTOD))
     .-DOMEGA*THETA2+180.
    2 CONTINUE
      RETURN
      END
```

```
E.LLIST                                                                                       28 JAN 72  15:02:42  CYCLE:  0
    1.@RUN    DPNLST,DPN,EREP
    2.@HDG    L LISTING OF MAIN ROUTINE      UN SAKOLOSKY
    3.@SYM   PRINT$,,TPR5
    4.@ASG,A    EREP
    5.@FOR,SL    EREP.MAIN,MAIN
```

```
E.LONRAN                                                  01 FEB 72  12:29:09  CYCLE:  0
      SUBROUTINE LONRAN(THETA,FOV,DOMN,DOMP)
        DATA SINI/.76604443/COSI/.64278761/RATIO/1.0682359/
     .   RTOD/57.2957795/E/.065871629/
        FOVR=FOV/RTOD
        RHO=ASIN(RATIO*SIN(FOVR))-FOVR
        COSRHO=COS(RHO)
        SINRHO=SIN(RHO)
        A=COSRHO*COSI
        B=SINRHO*SINI
        C=COSRHO*SINI
        D=SINRHO*COSI
      IF(THETA.GT.270..AND.THETA.LE.360.)THETA=THETA-360.
        THETAR=THETA/RTOD
        STHETA=SIN(THETAR)
        F=COSRHO*COS(THETAR)
        G=E*THETAR
        PHI=ASIN(STHETA*SINI)*RTOD
        XLAM=(ATAN2(STHETA*COSI,COS(THETAR))-G)*RTOD
        CALL LATLON(PHI,XLAM,OMGOA,OMGOD)
        PHIR=ASIN(C*STHETA-D)*RTOD
        XLAMR=(ATAN2((A*STHETA+B),F)-G)*RTOD
        CALL LATLON(PHIR,XLAMR,OMGRA,OMGRD)
        PHIL=ASIN(C*STHETA+D)*RTOD
        XLAML=(ATAN2((A*STHETA-B),F)-G)*RTOD
        CALL LATLON(PHIL,XLAML,OMGLA,OMGLD)
       DLA=OMGLA-OMGOA
       DRA=OMGRA-OMGOA
       DLD=OMGLD-OMGOD
       DRD=OMGRD-OMGOD
       IF(THETA.GT.-90..AND.THETA.LE.90.) GO TO 1
       IF(THETA.GT.90..AND.THETA.LE.270.) GO TO 2
    1  DOMN=DLA
       DOMP=DRA
      RETURN
    2  IF(DLD.LT.0.)DLD=DLD+360.
       DOMN=DRD
       DOMP=DLD
       RETURN
       END
```

```
E.MAIN                                                    08 MAR 72  01:48:39  CYCLE:  0
C
C
C
      INTEGER OPTION, SUBOPT
      INCLUDE BLK1,LIST
      INCLUDE BLK2,LIST
      INCLUDE BLK3,LIST
      INCLUDE BLK4,LIST
      INCLUDE BLK5,LIST
      COMMON/PRNT/IPRINT
      DATA IBLANK/' '/
      NAMELIST/MAX/ARCPAS,MCATEG,THMIN,THMAX
      NAMELIST/CONTRL/APASDE,MCATDE,THMIDE,THMADE,IDEVIC,MISION,
     .          IERSAV,IEROLD,PERWT,ARCMIN
C
C   INITIALIZE DATA, SET UP COMMON, READ FILES
C
      IPFLG = 0
      CALL CINOTE(IPFLG)
      ICHFLG = 0
      CALL COMMON(ICHFLG)
C
C   READ NEXT OPTION
C
    1 WRITE(6,1000)
      LETTER = IBLANK
      READ(5,2000) OPTION
      IF(OPTION .EQ. 'STOP' .OR. OPTION .EQ. 'HALT') GO TO 900
      FLD(0,6,LETTER) = FLD(0,6,OPTION)
      IF(LETTER .EQ. 'E') GO TO 100
      IF(LETTER .EQ. 'P') GO TO 200
      IF(LETTER .EQ. 'M') GO TO 300
      IF(LETTER .EQ. 'A') GO TO 400
      IF(LETTER .EQ. 'C') GO TO 500
      IF(LETTER .EQ. 'S') GO TO 600
      WRITE(6,2050) OPTION
      GO TO 1
C
C   EVALUATE REVOLUTION
C
  100 WRITE(6,2075)
      READ(5,2090) IREV
      IF(IREV .GE. TRAJ(1,1) .AND. IREV .LE. TRAJ(1,ITOTRJ)) GO TO 110
      WRITE(6,3000) IREV
      GO TO 1
  110 DO 120 J = 1, ITOTRJ
      IF(TRAJ(1,J) .GT. IREV) GO TO 125
  120 IF(IREV .EQ. TRAJ(1,J)) GOTO 130
  125 WRITE(6,3010) IREV
      GO TO 1
  130 CALL FOSCAN
      GO TO 1
C
C   PRINT OPTION
C
  200 WRITE(6,3020)
```

```
      READ(5,2000) SUBOPT
      IF(SUBOPT .NE. 'REVSUM') GO TO 205
      WRITE(6,3060) THSTRT,THEND
      READ(5,2000) IANS
      IF(IANS .NE. 'NO') GO TO 202
      WRITE(6,3065)
      READ(5,3070) THSTRT
      WRITE(6,3075)
      READ(5,3070) THEND
 202  CALL PSCRMC
      GO TO 1
C
 205  IF(SUBOPT .NE. 'PRPSUM') GO TO 210
      WRITE(6,3040)
      READ(5,2090) IPROP
      IHDG = 1
      IFOPRN = 1
      CALL PPROP(IHDG,IFOPRN,IPROP)
      GO TO 1
 210  IF(SUBOPT .NE. 'ERESUM') GO TO 220
      IHDG = 1
      IFOPRN = 0
      DO 215 J=1,NUMPRO
      IPROP = PRLIST(J)
      CALL PPROP(IHDG,IFOPRN,IPROP)
      IPF = IPFLG
      IF(IPF .EQ. 1) GO TO 216
 215  IHDG = 0
      GO TO 1
 216  IPFLG = 0
      READ(5,2000) IANS
      GO TO 1
 220  IF(SUBOPT .NE. 'PASSUM') GO TO 230
      CALL PPASS
      GO TO 1
 230  IF(SUBOPT .NE. 'NOWORK') GO TO 240
      IWORK = 0
      CALL NONDON
      GO TO 1
 240  IF(SUBOPT .NE. 'ALDONE') GO TO 250
      IWORK = 1
      CALL ALLDON
      GO TO 1
 250  IF(SUBOPT .NE. 'MISSUM') GO TO 260
      CALL PTSCRM
      GO TO 1
 260  IF(SUBOPT .NE. 'PRGSUM') GO TO 265
      CALL PTSCRP
      GO TO 1
 265  IF(SUBOPT .NE. 'PSCORE') GO TO 270
      WRITE(6,4000) ((SCORE(J,K),J = 1,6),K = 1,NSCORE)
      GO TO 1
 270  WRITE(6,3050) SUBOPT
      GO TO 1
C
C   MAXAGT OPTION
C
```

```
 300  ARCPAS = APASDE
      MCATEG = MCATDE
      THMIN = THMIDE
      THMAX = THMADE
      IPRINT = 0
      WRITE(6,3085) ARCPAS,MCATEG,THMIN,THMAX
      READ(5,2000) IANS
      IF(IANS .NE. 'NO') GO TO 310
      WRITE(6,3100)
      READ(5,MAX)
 310  CALL THEMAX(ARCPAS,MCATEG,THMAX,THMIN,WGTMXC,WGTMXU)
      GO TO 1
C
C   ACCEPT OPTION
C
 400  WRITE(6,3120) IREV, PERWT
      READ(5,2000) IANS
      IF(IANS .EQ. 'NO') GO TO 1
      WRITE(6,3060) THSTRT,THEND
      READ(5,2000) IANS
      IF(IANS .NE. 'NO') GO TO 410
      WRITE(6,3065)
      READ(5,3070) THSTRT
      WRITE(6,3075)
      READ(5,3070) THEND
 410  CALL WGTADD
      GO TO 1
C
C   CHANGE OPTION
C
 500  WRITE(6,3140)
      READ(5,2000) IANS
      IF(IANS .EQ. 'NO') GO TO 510
      WRITE(6,3130) APASDE,MCATDE,MISION,THMIDE,THMADE,PERWT,
     . IERSAV,IEROLD,ILEVIC,ARCMIN
 510  MISSAV = MISION
      WRITE(6,3150)
      READ(5,CONTRL)
      IF(MISION .EQ. MISSAV) GO TO 1
      ICHFLG = 1
      CALL COMMON (ICHFLG)
      GO TO 1
C
C   SCAN OPTION
C
 600  WRITE(6,3160)
      READ(5,2090) IFDAY
      WRITE(6,3170)
      READ(5,2090) ILDAY
      CALL MASCAN(IFDAY,ILDAY)
      GO TO 1
C
C   HALT OPTION
C
 900  IF(IERSAV .GT. 0) GO TO 910
      WRITE(6,3180)
      READ(5,2000) IANS
      IF(IANS .NE. 'NO') CALL EXIT
      WRITE(6,3190)
      READ(5,2090) IERSAV
  910 REWIND IERSAV
      WRITE(IERSAV) TOTSCR,PASDAT,OMECOV,PASS, IPASS
      WRITE(6,3200) IERSAV
      CALL EXIT
C
C   FORMATS
C
 1000 FORMAT(/T2,'OPTION = ')
 2000 FORMAT(A6)
 2050 FORMAT(T2,'OPTION ',A6,' IS NOT VALID.')
 2075 FORMAT(/T2,'REV NO.?      (I4 FORMAT)')
 2090 FORMAT(I4)
 3000 FORMAT(T2,'REV ',I4,' IS NOT IN THIS MISSION.')
 3010 FORMAT(T2,'REV ',I4,' NOT EVALUATED ',
     . 'BECAUSE ABSOLUTE BETA GREATER THAN 50')
 3020 FORMAT(T2,'PRINT OPTION = ')
 3040 FORMAT(/T2,'TYPE PROPOSAL NUMBER      (I4 FORMAT)')
 3050 FORMAT(T2,'OPTION ',A6,' NOT AVAILABLE')
 3060 FORMAT(/T2,'THSTRT = ',F9.4,5X,'THEND = ',F9.4,5X,'O.K.?')
 3065 FORMAT(/T2,'THSTRT = ')
 3070 FORMAT(F12.6)
 3075 FORMAT(/T2,'THEND = ')
 3085 FORMAT(/T2,'ARCPAS = ',F9.4,5X,'MCATEG = ',I3/
     . T2,'THMIN =  ',F9.4,5X,'THMAX = ',F9.4,5X,'O.K.?')
 3100 FORMAT(/T2,'SUPPLY VALUES, NAMELIST MAX')
 3120 FORMAT(/T2,'ACCEPT REV = ',I4,', PERWT = ',F7.3,'  OK?')
 3130 FORMAT(/T2,'CURRENT VALUES OF CONTROL VARIABLES ARE:'/
     . T2,'APASDE = ',F9.4,T22,'MCATDE = ',I3,
     . T42,'MISION = ',I3/
     . T2,'THMIDE = ',F9.4,T22,'THMADE = ',F9.4,
     . T42,'PERWT = ',F7.3/
     . T2,'IERSAV = ',I3,T22,'IEROLD = ',I3,T42,'IDEVIC = ',A6/
     . T2,'ARCMIN = ',F6.3)
 3140 FORMAT(/T2,'PRINT CONTROL DATA ?')
 3150 FORMAT(/T2,'ENTER CHANGED PARAMETERS, NAMELIST CONTRL')
 3160 FORMAT(/T2,'FIRST DAY OF SCAN = ?     (I4 FORMAT)')
 3170 FORMAT(/T2,'LAST DAY OF SCAN =  ?     (I4 FORMAT)')
 3180 FORMAT(/T2,'PROPOSAL ACCOMPLISHMENTS WILL NOT BE SAVED, O.K.?')
 3190 FORMAT(/T2,'SAVE ON UNIT NUMBER ?    (TYPE ANSWER IN I4 FORMAT)')
 3200 FORMAT(/T2,'PROPOSAL ACCOMPLISHMENTS HAVE BEEN WRITTEN ',
     . 'ON UNIT ',I4)
 4000 FORMAT(/T2,'EREP PTR',T13,'THETA S',T23,'THETA E',T33,
     . 'WGT UNC',T43,'WGT CON',T53,'CATEG'/
     . 100(T2,6(F8.2,2X)/))
      END
```

```
E.MASCAN                                                  06 MAR 72  15:17:38  CYCLE:  0
      SUBROUTINE MASCAN(IFDAY,ILDAY)
C
C  PURPOSE:   EXAMINE EVERY ORBIT FROM IFDAY THROUGH ILDAY
C             AND DETERMINE WHICH N ACCOMPLISH THE MOST IN TERMS
C             OF F.O. WEIGHTS.
C
C
      INCLUDE BLK1,LIST
      INCLUDE BLK2,LIST
      INCLUDE BLK5,LIST
C
C
      NAMELIST/INPUT/APASDE,MCATDE,THMIDE,THMADE
      DIMENSION FLY(5,700)
      COMMON/SCAN/ISCAN
      COMMON/PRNT/IPRINT
C
C
      IPRINT = 1
      WMAX = 0
      IFLY = 0
      NOMEGA = 0
      IBETA = 0
      WRITE(6,4000)
      READ(5,5150) IANS
      IF(IANS .NE. 'NO') GO TO 20
      NOMEGA = 1
      GO TO 90
   20 WRITE(6,4100)
      READ(5,4200) OMW
      WRITE(6,4250)
      READ(5,4200) OME
      IF(OMW .LT. 0) OMW = OMW + 360
      IF(OME .LT. 0) OME = OME + 360
      IF(OME .LT. OMW) OME = OME + 360
   90 WRITE(6,5000)
      READ(5,5050) N
      WRITE(6,5100) APASDE,MCATDE,THMIDE,THMADE
      READ(5,5150) IANS
      IF(IANS .NE. 'NO') GO TO 100
      WRITE(6,5200)
      READ(5,INPUT)
  100 START = IFDAY
      XEND = 1 + ILDAY
      DO 500 I = 1,ITOTRJ
      LINE = I
      IF(TRAJ(4,LINE) .LT. START) GO TO 500
      IF(TRAJ(4,LINE) .GT. XEND) GO TO 510
      IF( I .GE. NBETA1 .AND. I .LE. NBETA2) GO TO 120
      GO TO 130
  120 IF(IBETA .EQ. 1) GO TO 500
      IBETA = 1
      WRITE(6,6000)
      GO TO 500
  130 IF(NOMEGA .EQ. 1) GO TO 400
      IF(TRAJ(2,LINE) .LT. 0) TRAJ(2,LINE) = TRAJ(2,LINE) + 360
```

C.3

```
      IF(TRAJ(2,LINE) .GE. OMW .AND. TRAJ(2,LINE) .LE. OME)
     . GO TO 400
      IF(OME .GE. 360 .AND. TRAJ(2,LINE) + 360 .GE. OMW .AND.
     . TRAJ(2,LINE) + 360 .LE. OME) GO TO 400
      GO TO 500
  400 IREV = TRAJ(1,LINE)
      ISCAN = 1
      CALL FOSCAN
      CALL THFMAX(APASDE,MCATDE,THMADE,THMIDE,WGTMXC,WGTMXU)
      IFLY = IFLY + 1
      FLY(1,IFLY) = TRAJ(1,LINE)
      FLY(2,IFLY) = THSTRT
      FLY(3,IFLY) = THEND
      FLY(4,IFLY) = WGTMXC
      FLY(5,IFLY) = WGTMXU
  500 CONTINUF
C
C
  510 IFIRST = IFDAY
      ILAST = ILDAY
      WRITE(6,5250) N, IFIRST, ILAST
      DO 700 I = 1,N
      WMAX = 0
      DO 600 J = 1,IFLY
      IF(WMAX .LT. FLY(4,J)) IMAX = J
  600 WMAX = AMAX1(WMAX,FLY(4,J))
      NREV= FLY(1,IMAX)
      WRITE(6,5300) NREV, (FLY(K,IMAX),K= 2,5)
      FLY(4,IMAX) = 0
  700 CONTINUE
      IPRINT = 0
      RETURN
C
C   FORMATS
C
 4000 FORMAT(/T2,'OMEGA LIMITS DESIRED?')
 4100 FORMAT(T2,'TYPE IN WESTERN OMEGA LIMIT.')
 4200 FORMAT(F8.3)
 4250 FORMAT(T2,'TYPE IN EASTERN OMEGA LIMIT.')
 5000 FORMAT(/T2,'INPUT NUMBER OF ORBITS TO BE RETURNED      ',
     . '(I2 FORMAT)')
 5050 FORMAT(I2)
 5100 FORMAT(T2,'CURRENT PARAMETER VALUES ARE:    APASDE = ',F8.3/
     . T2,'MCATDE = ',I3,5X,'THMIDE = ',F8.3,5X,
     . 'THMADE = ',F8.3,'      OK?')
 5150 FORMAT(A6)
 5200 FORMAT(T2,'INPUT CHANGED PARAMETERS, NAMELIST INPUT')
 5250 FORMAT(/T2,'THE ',I2,' MAXIMUM WEIGHT REVS FROM DAYS ',
     . I4,' THROUGH ',I4/
     . T3,'REV',T10,'THSTRT',T24,'THEND',T36,
     . 'WGT CON',T47,' GT UNCON'/)
 5300 FORMAT(T2,I4,T10,F8.3,T22,F8.3,T34,F9.3,T47,F9.3)
 6000 FORMAT(/T2,'DAYS OF INTEREST INCLUDE PERIOD WHEN ABSOLUTE ',
     . 'BETA GREATER THAN 50')
      END
```

C3

```
E.MISION
            DIMENSION TRAJ(5,700)
            SL2STR = 120.708
            SL2END = 148.273
            SL3STR = 208.232
            SL3END = 264.217
            SL4STR = 304.604
            SL4END = 361.563
            IFILE = 2
            IMIS = 1
            START = SL2STR
            FINISH = SL2END
            WRITE(6,1050) IFILE
C
      50    READ(9, END = 200) NREV,DUM,XLON,BETSUN,DIJA73,THE20U,THE20D
            IF(DIJA73 .LT. START) GO TO 50
            IF(DIJA73 .GT. FINISH) GO TO 200
            IF(ABS(BETSUN) .LT. 50) GO TO 60
            IF(NBETA1 .NE. 0) GO TO 50
            WRITE(6,3000)
            NBETA1 = NREV
            GO TO 50
      60    IF(NBETA2 .NE. 0 .OR. NBETA1 .EQ. 0) GO TO 70
            NBETA2 = NREV - 1
      70    THNOON = (THE20U+THE20D)/2
            ICOUNT = ICOUNT + 1
            IF(ICOUNT .LE. 111) GO TO 100
            ICOUNT = 1
            WRITE(6,1050) IFILE
     100    WRITE(6,2000) NREV,XLON,BETSUN,DIJA73,THNOON
            I = I +1
            TRAJ(1,I) = NREV
            TRAJ(2,I) = XLON
            TRAJ(3,I) = BETSUN
            TRAJ(4,I) = DIJA73
            TRAJ(5,I) = THNOON
            GO TO 50
     200    ICOUNT = 0
            ITOTRJ = I
            WRITE(IFILE) ITOTRJ,NBETA1,NBETA2,TRAJ
            DO 220  K = 1,ITOTRJ
            DO 220 J = 1,5
     220    TRAJ(J,K) = 0
            I = 0
            NBETA1 = 0
            NBETA2 = 0
            GO TO (250,300,350), IMIS
     250    IFILE = 3
            IMIS = 2
            START = SL3STR
            FINISH = SL3END
            WRITE(6,1050) IFILE
            GO TO 50
     300    IFILE = 4
            IMIS = 3
            START = SL4STR
            FINISH = SL4END
      WRITE(6,1050) IFILE
      GO TO 50
  350 CONTINUE
C
 1050 FORMAT(1H1/T2,'SL',I1,T8,'NREV',T18,'XLON',T29,'BETSUN',T41,
     .'01JA73',T54,'THNOON')
 2000 FORMAT(T8,I4,T15,F9.4,T28,F8.4,T40,F9.4,T53,F8.4)
 3000 FORMAT(T2,'ABSOLUTE BETA GREATER THAN 50')
      END
```

```
E.MSORT                                                                     20 MAR 72  09:46:40  CYCLE:  0
C
C         THIS ROUTINE SORTS THE EREP ARRAY BY MISSION
C         CATEGORY, REARRANGES THE ARRAY AND OUTPUTS
C         THE ARRAY IN BINARY ALONG WITH THE POINTERS TO
C         THE LIMITS OF EACH CATEGORY.
C
          INTEGER SL2(2), SL3(2), SL4(2), SEASON
          DIMENSION MISION(7,400), IMIS(7), NONE(50),
        . EREP(15,1000), EREP2(15,1000), IERPNT(7,2), IHIER(7)
          DATA     IHIER/2,23,24,234,3,34,4/
          DATA     SL2/5,5/, SL3/7,9/, SL4/10,12/
          NAMELIST/INEREP/EREP
C
C
          READ(5,INEREP)
     10   J=J+1
          IF(EREP(1,J) .LT. 0) GO TO 500
          IYR = 0
          ISL2 = 0
          ISL3 = 0
          ISL4 = 0
          SEASON=EREP(6,J)
          ISTART=SEASON/100
          IEND=MOD(SEASON,100)
          IF(IEND .LT. ISTART) IYR=1
C
C
          IF(ISTART .GE. SL2(1) .AND. ISTART .LE. SL2(2)
        . .OR. IEND .GE. SL2(1) .AND. IEND .LE. SL2(2)
        . .OR. ISTART .LT. SL2(1) .AND. IEND .GE. SL2(2)) GO TO 20
          IF(IYR .EQ. 0) GO TO 30
          IF(ISTART .LE. SL2(2) .AND. IEND .LE. SL2(2)
        . .OR. ISTART .GE. SL2(2) .AND. IEND .GE. SL2(1)) GO TO 20
          GO TO 30
     20   ISL2=1
     30   IF(ISTART .GE. SL3(1) .AND. ISTART .LE. SL3(2)
        . .OR. IEND .GE. SL3(1) .AND. IEND .LE.SL3(2)
        . .OR. ISTART .LF. SL3(1) .AND. IEND .GE. SL3(2)) GO TO 40
          IF(IYR .EQ. 0) GO TO 50
          IF(ISTART .LE. SL3(2) .AND. IEND .LE. SL3(2)
        . .OR. ISTART .GE. SL3(2) .AND. IEND .GE. SL3(1)) GOTO 40
          GO TO 50
     40   ISL3=1
     50   IF(ISTART .GE. SL4(1) .AND. ISTART .LE. SL4(2)
        . .OR. IEND .GE. SL4(1) .AND. IEND .LE. SL4(2)
        . .OR. ISTART .LE. SL4(1) .AND. IEND .GE. SL4(2)) GO TO 60
          IF(IYR .EQ. 0) GO TO 70
          IF(ISTART .LE. SL4(2) .AND. IEND .LE. SL4(2)
        . .OR. ISTART .GE. SL4(2) .AND. IEND .GE. SL4(1)) GO TO 60
          GO TO 70
     60   ISL4=1
C
C        DETERMINE FLIGHT CATAGORY FOR FO
C
     70  IF(ISL2 .EQ. 1 .AND. ISL3 .EQ. 1 .AND. ISL4 .EQ. 1) GO TO 80
         IF(ISL2 .EQ. 1 .AND. ISL3 .EQ. 1 .AND. ISL4 .EQ. 0) GO TO 90
```

```
      IF(ISL2 .EQ. 1 .AND. ISL3 .EQ. 0 .AND. ISL4 .EQ. 1) GO TO 150
      IF(ISL2 .EQ. 1 .AND. ISL3 .EQ. 0 .AND. ISL4 .EQ. 0) GO TO 170
      IF(ISL3 .EQ. 1 .AND. ISL4 .EQ. 1) GO TO 140
      IF(ISL3 .EQ. 1 .AND. ISL4 .EQ. 0) GO TO 190
      IF(ISL4 .EQ. 1) GO TO 210
      GO TO 230
   80 K = 4
      GO TO 240
   90 K = 2
      GO TO 240
  140 K = 6
      GO TO 240
  150 K = 3
      GO TO 240
  170 K = 1
      GO TO 240
  190 K = 5
      GO TO 240
  210 K = 7
      GO TO 240
  230 IN = IN + 1
      NONE(IN) = J
      GO TO 10
  240 IMIS(K) = IMIS(K) + 1
      NUM = IMIS(K)
      MISION(K,NUM) = J
      GO TO 10
  500 DO 550 N = 1,7
      NUM = IMIS(N)
      IF(NUM .NE. 0) GO TO 505
      WRITE(6,4000) IHIER(N)
      GO TO 550
  505 IF(N .EQ. 1) GO TO 510
      IERPNT(N,1) = IERPNT(N-1,2) + 1
      IERPNT(N,2) = IERFNT(N,1) + NUM - 1
      GO TO 520
  510 IERPNT(I,1) = 1
      IERPNT(1,2) = NUM
  520 N1 = IERPNT(N,1)
      N2 = IERPNT(N,2)
      DO 530  I = N1,N2
      NPT = MISION(N,I-N1+1)
      DO 530  J = 1,15
  530 EREP2(J,I) = EREP(J,NPT)
  550 CONTINUE
C
C     PRINT NEW EREP ARRAY
C
      DO 600 J = 1,7
      I1 = IERPNT(J,1)
      I2 = IERPNT(J,2)
  600 WRITE(6,1000) IMIS(J), IHIER(J), (IERPNT(J,K), K = 1,2),
     . ((K,(EREP2(I,K), I = 1,15)),K = I1,I2)
      WRITE(6,2000)
      DO 650 J = 1,IN
      NPT = NONE(J)
  650 WRITE(6,3000) EREP(1,NPT), EREP(2,NPT)
```

```
          WRITE(10) IERPNT,EREP2
C
C
    1000  FORMAT(1H1,T2,I3,' F.O.S ARE IN CATEGORY ',I3/
        . T2,'THEY COMPRISE EREP ENTRIES ',I4,' TO ',I4////
        . T3,'NUM',T7,'PROP  FO  SITE D  SENSRS SEASN PASS',
        . T44,'NO    LIGHT    CLOUD    SNOW     WEIGHT'/
        . 500(T2,I4,1X,F5.0,1X,F3.0,1X,F4.0,1X,F2.0,1X,
        . F6.0,1X,F5.0,F6.0,1X,F3.0,F4.0,1X,F4.0,1X,
        . F3.0,1X,F4.0,1X,F3.0,1X,F4.0,1X,F9.5/))
    2000  FORMAT(1H1///T2,'THE FOLLOWING ARE INCLUDED IN NO CATEGORY:'//
        . T10,'PROP',5X,'F.O.')
    3000  FORMAT(T10,F5.0,5X,F3.0)
    4000  FORMAT(/////T2,'*****THERE ARE NO ENTRIES IN MISSION CATEGORY ',
        . I3,' *****')
          END
```

```
E.NONDON                                                       06 MAR 72  01:51:24  CYCLE:  0
      SUBROUTINE NONDON
C
C   PURPOSE:   LIST THE F.O.'S WHICH CAN BE DONE ON THE
C              CURRENT MISSION FOR WHICH ACCUMULATED
C              SCORE = 0 (IWORK = 0)
C
      INCLUDE BLK1,LIST
      INCLUDE BLK2,LIST
      INCLUDE BLK3,LIST
      INCLUDE BLK4,LIST
      DIMENSION PRINT(5,2), IPR(5)
      EQUIVALENCE (PRINT(1,2),IPR(1))
      IPFLG = 0
      DO 10 J = 1,5
      DO 10 K = 1,2
   10 PRINT(J,K) = 0
      CALL CINOTE(IPFLG)
      WRITE(6,1000)
      DO 100 I = 1,4
      I1 = IEREP(I,1)
      I2 = IEREP(I,2)
      DO 100 J = I1,I2
      IF(TOTSCR(J) .GT. 0) GO TO 100
      PRINT(I,1) = PRINT(I,1) + EREP(15,J)
      IPR(I) = IPR(I) + 1
      IF(IPFLG .NE. 0) GO TO 100
      NPRO = EREP(1,J)
      NFO = EREP(2,J)
      WRITE(6,1050) NPRO,NFO,MCAT(I),EREP(15,J)
  100 CONTINUE
      WRITE(6,1075)
      DO 200 I = 1,4
      PRINT(5,1) = PRINT(5,1)+PRINT(I,1)
  200 IPR(5) = IPR(5) + IPR(I)
      WRITE(6,2000) ((MCAT(K),(PRINT(K,L),L = 1,2)),K = 1,4),
     . (PRINT(5,L),L = 1,2)
      IPFLG = 0
      RETURN
 1000 FORMAT(//T2,'F.O.S ON WHICH NO WORK HAS BEEN DONE'/
     . T2,'PROP',T8,'FO',T13,'MCAT',T20,'POS SCR')
 1050 FORMAT(T2,I4,T8,I2,T13,I3,T20,F6.2)
 1075 FORMAT(//T2,'SUMMARY'/T2,'MISSION CAT',T16,
     . 'TOT POS SCR',T30,'NO OF FOS')
 2000 FORMAT(4(T3,I3,T18,F7.1,T32,I3/),
     . T3,'TOTAL',T18,F7.1,T31,I4)
      END
```

```
E.PASBAD                                                      08 MAR 72  01:51:27  CYCLE:  0
      SUBROUTINE PASBAD(TSTART,TEND,THSTRT,THEND,IGOOD)
C
C  A PASS OVER A SITE WILL BE CALLED GOOD IF AT LEAST ONE-HALF THE
C  THETA RANGE OF THE SITE IS COVERED OR ARCMIN DEGREES OF THE
C  ZLV PASS IS SPENT OVER THE SITE.
C
      INCLUDE BLK1,LIST
C
C
      IF(THSTRT .LT. TSTART) GO TO 100
      IF(THSTRT .GE. TEND) GO TO 180
C
C  PARTIAL COVERAGE AT LEAST
C
      IF(THEND .GE. TEND) THDUM = TEND - THSTRT
      IF(THEND .LT. TEND) THDUM = THEND - THSTRT
      IF(THDUM .GE. ARCMIN) GO TO 190
      IF(THDUM .GE. ((TEND - TSTART)/2)) GO TO 190
      GO TO 180
C
  100 IF(THEND .LT. TSTART) GO TO 180
      IF(THEND .GT. TEND) GO TO 190
      IF(THEND .GE. (THSTRT + ARCMIN)) GO TO 190
      IF((THEND - TSTART) .GE. ((TEND - TSTART)/2)) GO TO 190
C
  180 IGOOD = 0
      RETURN
  190 IGOOD = 1
      RETURN
      END
```

```
E.PPASS                                              08 MAR 72  01:51:31  CYCLE:  0
           SUBROUTINE PPASS
 C
 C    PURPOSE:   PRINT A SUMMARY OF THE PASS DATA ACCEPTED SO FAR
 C
           INCLUDE BLK3,LIST
           INCLUDE BLK4,LIST
           WRITE(6,1000) IPASS
           DO 100 I = 1,IPASS
           CALL TIMDAT(PASS(3,I),MO,IDAY,HR)
           HR = HR + PASS(4,I) * ORBPER/21600.
           IF(HR .GT. 24) GO TO 20
           IF(HR .GE. 0) GO TO 30
           HR = HR + 24.
           IDAY = IDAY - 1
           GO TO 30
     20    HR = HR - 24
           IDAY = IDAY + 1
     30    IREV = PASS(1,I)
    100    WRITE(6,2000) IREV, PASS(2,I), MO, IDAY, HR,
         . (PASS(K,I),K = 4,7)
           RETURN
   1000    FORMAT(//T2,I3,T6,'PASSES HAVE BEEN ACCEPTED'//
         . T19,'PASS STARTS',T52,'SUN ELEVATION'/T2,'REV NO',T11,
         . 'OMEGA',T19,'MO DAY  HR',T36,'THSTRT  THEND   START    END')
   2000    FORMAT(T3,I4,T11,F6.2,T19,I2,T23,I2,T27,F7.4,T35,
         . F6.2,T44,F6.2,T52,F5.2,T60,F5.2)
           END
```

```
E.PPROP                                                         08 MAR 72  01:51:35  CYCLE:  0
      SUBROUTINE PPROP(IHDG,IFOPRN,IPROP)
C
C  PURPOSE:    PRINT CONDITIONAL, UNCONDITIONAL, AND ACCUMULATED
C              SCORES FOR A GIVEN PROPOSAL, IPROP.  IF IFOPRN = 1,
C              LIST EACH F.O. SEPARATELY.  OTHERWISE LIST ONLY TOTALS.
C
      INCLUDE BLK1,LIST
      INCLUDE BLK2,LIST
      INCLUDE BLK3,LIST
      INCLUDE BLK4,LIST
      INCLUDE BLK5,LIST
      DIMENSION PRINT(2,4)
      IF(IHDG .NE. 1) GO TO 20
      IHDG = 0
      WRITE(6,1000) IREV
   20 DO 30 J = 1,2
      DO 30 K = 1,4
   30 PRINT(J,K) = 0
      IGOOD = 0
      IER = IERPNT(7,2)
      DO 200 I = 1,IER
      IF(EREP(1,I) .NE. IPROP) GO TO 200
      IGOOD = 1
      DO 150 J = 1,NSCORE
      IF(SCORE(1,J) .NE. I) GO TO 150
      PRINT(1,1) = SCORE(5,J)
      PRINT(1,2) = SCORE(4,J)
      GO TO 160
  150 CONTINUE
      PRINT(1,1) = 0
      PRINT(1,2) = 0
  160 IF(IFOPRN .NE. 1) GO TO 180
      IFO = EREP(2,I)
      WRITE(6,2000) IPROP,IFO,PRINT(1,1),PRINT(1,2),
     . TOTSCR(I), EREP(15,I)
  180 PRINT(2,1) = PRINT(2,1) + PRINT(1,1)
      PRINT(2,2) = PRINT(2,2) + PRINT(1,2)
      PRINT(2,3) = PRINT(2,3) + TOTSCR(I)
      PRINT(2,4) = PRINT(2,4) + EREP(15,I)
  200 CONTINUE
      IPRO = IPROP
      IF(IGOOD .NE. 0) GO TO 250
      WRITE(6,2050) IPRO
      RETURN
  250 WRITE(6,3000) IPRO, (PRINT(2,K),K = 1,4)
      RETURN
 1000 FORMAT(//T2,'PROPOSAL SUMMARY, REV = ',I4/
     . T2,'PROP',T10,'FO',T16,'PASS COND',T27,'PASS UNCOND',
     . T41,'ACCUM',T49,'TOT POS')
 2000 FORMAT(T2,I4,T10,I2,T17,F5.1,T30,F5.1,T41,F5.1,T50,F5.1)
 2050 FORMAT(/T2,'PROPOSAL ',I4,' IS NOT LISTED')
 3000 FORMAT(T2,I4,T8,'TOTAL',T17,F5.1,T30,F5.1,T41,F5.1,T50,F5.1)
      END
```

```
E.PSCRMC                                                   08 MAR 72  01:54:02  CYCLE:  0
           SUBROUTINE PSCRMC
C
C    PURPOSE:    PRINT SUMMARY OF CURRENT SCORE MATRIX BY CATEGORY.
C
           INCLUDE BLK1,LIST
           INCLUDE BLK2,LIST
           INCLUDE BLK3,LIST
           INCLUDE BLK4,LIST
           INCLUDE BLK5,LIST
           DIMENSION PRINT(5,5), NFO(5)
           IF(IDEVIC .NE. 'PRNTR') GO TO 50
           WRITE(6,2000)
           DO 20 J = 1,NSCORE
           NPT = SCORE(1,J)
           IPROP = EREP(1,NPT)
           IFO = EREP(2,NPT)
           TWEIGH = EREP(15,NPT)
           MCATEG = SCORE(6,J)
           CUR = TOTSCR(NPT)
      20   WRITE(6,2010) IPROP,IFO,(SCORE(K,J),K = 2,3),TWEIGH,
         . (SCORE(K,J), K = 4,5), MCATEG, CUR
C
C    ZERO OUT PRINT MATRIX
C
      50   DO 65 J = 1,5
           DO 60 K = 1,5
      60   PRINT(K,J) = 0
      65   NFO(J) = 0
           DO 100 I = 1,NSCORE
           NPT = SCORE(1,I)
           CALL PASBAD(SCORE(2,I),SCORE(3,I),THSTRT,THEND,IGOOD)
           IF(IGOOD .EQ. 0) GO TO 100
           K = 0
           DO 70 J = 1,3
           IF(SCORE(6,I) .EQ. MCAT(J)) K = J
      70   IF(K .NE. 0) GO TO 80
           K = 4
      80   PRINT(K,1) = PRINT(K,1) + SCORE(5,I)
           PRINT(K,2) = PRINT(K,2) + SCORE(4,I)
           PRINT(K,3) = PRINT(K,3) + TOTSCR(NPT)
           PRINT(K,4) = PRINT(K,4) + EREP(15,NPT)
     100   NFO(K) = NFO(K) + 1
           DO 300 J = 1,4
           DO 200 K = 1,4
     200   PRINT(5,J) = PRINT(5,J) + PRINT(K,J)
     300   NFO(5) = NFO(5) + NFO(J)
           WRITE(6,3000) IREV,THSTET,THEND,((MCAT(K),(PRINT(K,J),J = 1,4),
         . NFO(K)),K = 1,4), (PRINT(5,J), J = 1,4), NFO(5)
           RETURN
    2000   FORMAT(//T2,'PROP',T9,'FO',T14,'TSTRT',T23,
         . 'TEND',T30,'TWEIGH',T38,'WGT U',T46,'WGT C',
         . T53,'MCATEG',T60,'CURWGT')
    2010   FORMAT(T2,I4,T9,I2,T14,F5.1,T22,F5.1,T30,F5.1,
         . T38,F5.1,T46,F5.1,T54,I3,T60,F5.1)
    3000   FORMAT(//T2,'SUMMARY OF REV ',I3,', THSTRT = ',F5.1,
         . ', THEND = ',F5.1//T2,'MCATEG',T10,
     . 'WGT COND',T20,'WGT UNC',T29,'ACCUM',
     . T36,'TOT WGT POS',T50,'NO FOS'/
     . 4(T3,I3,T9,F7.1,T19,F7.1,T27,F7.1,T37,F7.1,
     . T52,I3/),T2,'TOTAL',T9,F7.1,T19,F7.1,
     . T27,F7.1,T37,F7.1,T51,I4)
       END
```

```
E.PTSCRM                                                    08 MAR 72  01:54:07  CYCLE:  0
           SUBROUTINE PTSCRM
C
C   PURPOSE:     PRINT OUT NO OF FO'S WORKED ON AND ACCUMULATED
C                SCORE FOR CURRENT MISSION--TOTALS AND BY MISSION
C                CATEGORY.  IF PRINTER, PRINT EACH F.O.
C
           INCLUDE BLK1,LIST
           INCLUDE BLK2,LIST
           INCLUDE BLK3,LIST
           INCLUDE BLK4,LIST
           DIMENSION PRINT(5,2), KNT(5,2)
           DO 10 J = 1,5
           DO 10 K = 1,2
           PRINT(J,K) = 0
     10    KNT(J,K) = 0
           DO 50 K = 1,4
           ILIM1 = IEREP(K,1)
           ILIM2 = IEREP(K,2)
           IF(IDEVIC .NE. 'PRNTR') GO TO 40
           WRITE(6,1000) MCAT(K)
           DO 30 I = ILIM1,ILIM2
           NPRO = EREP(1,I)
           NFO = EREP(2,I)
     30    WRITE(6,1050) NPRO,NFO,TOTSCR(I),EREP(15,I)
     40    DO 50 I = ILIM1,ILIM2
           PRINT(K,1) = PRINT(K,1) + TOTSCR(I)
           PRINT(K,2) = PRINT(K,2) + EREP(15,I)
           KNT(K,2) = KNT(K,2) + 1
           IF(TOTSCR(I) .LE. 0) GO TO 50
           KNT(K,1) = KNT(K,1) + 1
     50    CONTINUE
           DO 60 I = 1,4
           DO 60 J = 1,2
           PRINT(5,J) = PRINT(5,J) + PRINT(I,J)
     60    KNT(5,J) = KNT(5,J) + KNT(I,J)
           WRITE(6,2000) MISION
           WRITE(6,2050) ((MCAT(I),(PRINT(I,J),J = 1,2),
         • (KNT(I,J),J = 1,2)),I = 1,4),
         • (PRINT(5,J),J = 1,2),(KNT(5,J),J = 1,2)
           RETURN
   1000    FORMAT(//T2,'SUMMARY OF FOS FOR MISSION CATEGORY ',I3/
         • T2,'PROP',T9,'FC NO.',T17,'WGT',T22,'SAT.',
         • T28,'TOT WGT POS')
   1050    FORMAT(T2,I4,T11,I2,T19,F5.1,T31,F5.1)
   2000    FORMAT(//T2,'SUMMARY OF ACCUMULATED SCORE FOR SL',I1,' MISSION'/
         • T2,'MISSION CAT',T16,'WGT SAT',T25,'TOT WGT POS',
         • T38,'NON ZERO FOS',T52,'TOT FOS')
   2050    FORMAT(4(T4,I3,T15,F7.1,T26,F7.1,T42,I4,T54,I4/),
         • T4,'TOTAL',T15,F7.1,T26,F7.1,T42,I4,T54,I4)
           END
```

```
E.PTSCRP                                                        08 MAR 72  01:54:12  CYCLE:  0
          SUBROUTINE PTSCRP
C
C     PURPOSE:   SUMMARIZE ACCUMULATED FO WEIGHTS ACCOMPLISHED
C
          INCLUDE BLK1,LIST
          INCLUDE BLK2,LIST
          INCLUDE BLK3,LIST
          INCLUDE BLK4,LIST
          DIMENSION KNT(8,3), PRINT(8,2)
          DO 10 J = 1,8
          DO 5 K = 1,2
          PRINT(J,K) = 0
      5   KNT(J,K) = 0
     10   KNT(J,3) = 0
          KNT(1,1) = 2
          KNT(2,1) = 23
          KNT(3,1) = 24
          KNT(4,1) = 234
          KNT(5,1) = 3
          KNT(6,1) = 34
          KNT(7,1) = 4
          KNT(8,1) = 'TOTAL'
          DO 100 K = 1,7
          I1 = IERPNT(K,1)
          I2 = IERPNT(K,2)
C
          DO 100 I = I1,I2
          PRINT(K,1) = PRINT(K,1) + TOTSCR(I)
          PRINT(K,2) = PRINT(K,2) + EREP(15,I)
          KNT(K,3) = KNT(K,3) + 1
          IF(TOTSCR(I) .LE. 0) GO TO 100
          KNT(K,2) = KNT(K,2) + 1
    100   CONTINUE
          DO 200 I = 1,7
          DO 130 J = 1,2
    130   PRINT(8,J) = PRINT(8,J) + PRINT(I,J)
          DO 140 J = 2,3
    140   KNT(8,J) = KNT(8,J) + KNT(I,J)
    200   CONTINUE
          WRITE(6,1000)
          WRITE(6,2000) ((KNT(I,1),(PRINT(I,J),J=1,2),
        . (KNT(I,J),J=2,3)),I=1,8)
          RETURN
C
C         FORMATS
C
   1000   FORMAT(//T2,'FO WEIGHT ACCOMPLISHED SO FAR:'/
        . T2,'NCATEG',T11,'ACCUM WGT  TOT WT POS',
        . T35,'NON ZERO FOS',T50,'TOT FOS')
   2000   FORMAT(7(T3,I3,T11,F7.1,T22,F7.1,T39,I4,T51,I4/),
        . T2,A6,T11,F7.1,T22,F7.1,T39,I4,T51,I4)
          END
```

```
E.RECOMP                                                    11 FEB 72  08:36:25  CYCLE:  0
   @ RUN   OPNRE,OPNCOM,EREP,35,100
   @HDG    RECOMPILE ELEMENTS ON EREP FILE      ON SAKOLOSKY
   @SYM    PRINT$,,TPR5
   @ASG,AX EREP
   @FOR,S EREP.ALLDON,.ALLDON,.ALLDON
   @FOR,S EREP.COMMON,.COMMON,.COMMON
   @FOR,S EREP.FLYOVR,.FLYOVR,.FLYOVR
   @FOR,S EREP.FOSCAN,.FOSCAN,.FOSCAN
   @FOR,S EREP.MAIN,.MAIN,.MAIN
   @FOR,S EREP.MISION,.MISION,.MISION
   @FOR,S EREP.NONDON,.NONDON,.NONDON
   @FOR,S EREP.PASBAD,.PASBAD,.PASBAD
   @FOR,S EREP.PPASS,.PPASS,.PPASS
   @FOR,S EREP.PPROP,.PPROP,.PPROP
   @FOR,S EREP.PSCRMC,.PSCRMC,.PSCRMC
   @FOR,S EREP.PTSCRM,.PTSCRM,.PTSCRM
   @FOR,S EREP.PTSCRP,.PTSCRP,.PTSCRP
   @FOR,S EREP.THEMAX,.THEMAX,.THEMAX
   @FOR,S EREP.WGTADD,.WGTADD,.WGTADD
   @PACK   EREP.
   @PREP   EREP.
   @MAP,AL     EREP.ERMAP,.ERMAP
   @PREP       EREP.
   @PRT,T EREP.
```

```
E.SITES                                                22 MAR 72  15:55:21  CYCLE:  0
C
C
C  THIS PROGRAM READS A NAMELIST INPUT CONTAINING
C     EREP, AN ARRAY OF EREP SITE LAT AND LON VALUES WITH SITES
C           SEPARATED BY 9999.0,9999.0
C     EREPNO, AN ASSOCIATED ARRAY CONTAING SITE NUMBERS IN THE ORDER
C              IN WHICH THEY OCCUR IN EREP
C
      DIMENSION THOMEG(4,5500), EREP(2,5500), INDEX(2,600),
     . SITE(2,5000)
      INTEGER EREPNO(600)
      REAL INCL
      EQUIVALENCE (THOMEG(1,1), EREP(1,1))
      NAMELIST/INEREP/EREP,EREPNO
      NAMELIST/OUT2/NENTR,INDEX,THOMEG
C
C     ROTATION RATE OF THE EARTH  (DEG/HR)
      OMEGAE = 15.04104953
C
C
C     ORBITAL ROTATION RATE OF THE SATELLITE   (DEG/HR)
      OMEGAO = 360./1.55471699
C
C
C     REGRESSION RATE OF THE LINE OF NODES OF THE SATELLITE ORGIT (DEG/HR)
      REGRES = .21174988
C
      PI = 3.14159
      DTOR = PI/180.0
      RTOD = 180.0/PI
      EARTH = (OMEGAE + REGRES)/OMEGAO
      INCL=50.0
      RINCL=INCL*DTOR
      SINRIN=SIN(RINCL)
      RINCL = COS(RINCL)
C
C
      READ(5,INEREP)
C
      MAXNO = EREPNO(1)
   10 J=J+1
      MAXNO = MAXO(MAXNO,EREPNO(J))
   20 K=K+1
      IF(ABS(EREP(1,K)).GT. 90) GO TO 30
      L=L+1
      SITE(1,L)=EREP(1,K)
      SITE(2,L)=EREP(2,K)
      NPAIR=NPAIR+1
      GO TO 20
C
C  A COMPLETE SITE DEFINITION HAS BEEN READ
C
   30 JJ=EREPNO(J)
      INDEX(1,JJ)=L+1-NPAIR
      INDEX(2,JJ)=NPAIR
      NPAIR=0
      NENTR=NENTR+1
      I = INDEX(1,JJ)
      WRITE(6,7777) NENTR, EREPNO(J), ((SITE(N1,N2),N1 =1,2),
     . N2 = I,L)
 7777 FORMAT(//T2,I4,T10,I4,10(T20,2(F9.4,2X)/))
      IF(EREP(1,K) .LT. 0) GO TO 40
      GO TO 10
   40 WRITE(2) NENTR, INDEX,SITE
      WRITE(6,6666) NENTR, EREP(1,K), EREP(2,K),
     . SITE(1,L), SITE(2,L)
 6666 FORMAT(//T2,'NENTR = ',I4,2X,'EREP = ',2(F9.4,2X),
     . /T2,'SITE = '2(F9.4,2X))
C
C   COMPUTE THETA AND OMEGA, ASCENDING AND DESCENDING, FOR
C   EACH LAT AND LON
      DO 100 K=1,L
      QLAT=SITE(1,K)
      QLON=SITE(2,K)
      IF(QLON .LT. 0) QLON=360+QLON
      IF(ABS(QLAT) .LT. INCL) GO TO 50
      IF(QLAT .LT. 0) QLAT=-49.999
      IF(QLAT .GT. 0) QLAT=49.999
   50 THETAU=ASIN(SIN(QLAT*DTOR)/SINRIN)
      THETAD=PI-THETAU
      OMEGAU=QLON*DTOR-ATAN2(RINCL*SIN(THETAU),
     . COS(THETAU))+EARTH*THETAU
      OMEGAD=QLON*DTOR-ATAN2(RINCL*SIN(THETAD),
     . COS(THETAD))+EARTH*THETAD
      THETAU=THETAU*RTOD
      THETAD=THETAU*RTOD
      OMEGAU=OMEGAU*RTOD
      IF(OMEGAU .GE. 360) OMEGAU=AMOD(OMEGAU,360.)
      IF(OMEGAU .LT. 0) OMEGAU=OMEGAU+360
      OMEGAD=OMEGAD*RTOD
      IF(OMEGAD .GE. 360) OMEGAD=AMOD(OMEGAD,360.)
      IF(OMEGAD .LT. 0) OMEGAD=OMEGAD+360
      THOMEG(1,K)=THETAU
      THOMEG(2,K)=OMEGAU
      THOMEG(3,K)=THETAD
  100 THOMEG(4,K)=OMEGAD
      WRITE(3,OUT2)
      DO 200 K=1,MAXNO
      KK=K
      ISTART=INDEX(1,K)
      IEND=ISTART-1+INDEX(2,K)
  200 WRITE(6,1000) KK,(((SITE(J,M),J=1,2),
     . (THOMEG(J,M),J=1,4)),M=ISTART,IEND)
C
 1000 FORMAT(///T2,'SITE ',I3,T20,'LATITUDE',T33,'LONGITUDE',
     . T47,'THETAU',T60,'OMEGAU',T74,'THETAD',T87,'OMEGAD'//
     . 20(T21,F7.2,T34,F7.2,T46,F7.2,T59,F7.2,T73,F7.2,T86,F7.2/))
      END
```

```
E.SLOPE                                                        28 JAN 72  14:15:21  CYCLE:  0
          SUBROUTINE SLOPE(X1,Y1,X2,Y2,OM,THET)
          SLOPE = (Y2-Y1)/(X2-X1)
          YINTCP = Y1 - SLOPE * X1
          THET = SLOPE*OM+YINTCP
          RETURN
          END
```

```
E.STCOMP                                                   04 FEB 72  08:23:56  CYCLE:  0
@RUN    DPNCOM,OPN,EREP,15,100
 @HDG      COMPILE ELEMENTS ON EREP      ON SAKOLOSKY
 @SYM   PRINT$,,TPR5
 @ASG,AX EREP
 @PRT,T    EREP.
 @PDP,FL            EREP.COM,COM
 @EOF
 @HDG   SUBROUTINE ALLDON
 @FOR,S    EREP.ALLDON,ALLDON
 @HDG   SUBROUTINE COMMON
 @FOR,S    EREP.COMMON,COMMON
 @HDG   SUBROUTINE ELTHET
 @FOR,S    EREP.ELTHET,ELTHET
 @HDG   SUBROUTINE FLYOVR
 @FOR,S    EREP.FLYOVR,FLYOVR
 @HDG   SUBROUTINE FOSCAN
 @FOR,S    EREP.FOSCAN,FOSCAN
 @HDG   SUBROUTINE LATLON
 @FOR,S            EREP.LATLON,LATLON
 @HDG   SUBROUTINE LONRAN
 @FOR,S EREP.LONRAN,LONRAN
 @HDG   MAIN ROUTINE
 @FOR,S    EREP.MAIN,MAIN
 @HDG  MISION ROUTINE
 @FOR,S  EREP.MISION,MISION
 @HDG   SUBROUTINE NONDON
 @FOR,S    EREP.NONDON,NONDON
 @HDG   SUBROUTINE PASBAD
 @FOR,S    EREP.PASBAD,PASBAD
 @HDG   SUBROUTINE PPASS
 @FOR,S    EREP.PPASS,PPASS
 @HDG   SUBROUTINE PPROP
 @FOR,S    EREP.PPROP,PPROP
 @HDG   SUBROUTINE PSCRMC
 @FOR,S    EREP.PSCRMC,PSCRMC
 @HDG   SUBROUTINE PTSCRM
 @FOR,S    EREP.PTSCRM,PTSCRM
 @HDG   SUBROUTINE PTSCRP
 @FOR,S    EREP.PTSCRP,PTSCRP
 @HDG   SUBROUTINE SUNEL
 @FOR,S    EREP.SUNEL,SUNEL
 @HDG  SUBROUTINE THEMAX
 @FOR,S            EREP.THEMAX,THEMAX
 @HDG   SUBROUTINF TIMDAT
 @FOR,S    EREP.TIMDAT,TIMDAT
 @HDG      SUBROUTINE SLOPE
 @FOR,S            EREP.SLOPE,SLOPE
 @HDG   SUBROUTINE VROTAT
 @FOR,S    EREP.VROTAT,VROTAT
 @HDG   SUBROUTINE MGTADD
 @FOR,S    EREP.MGTADD,MGTADD
 @MAP,AL      EREP.ERMAP,ERMAP
```

```
E.SUNEL
      SUBROUTINE SUNEL(BETA,TNOON,THETA,EL,NOSUCH)
C
C
C  PURPOSE:  GIVEN BETA,AND THE VALUES OF THETA WHEN THE SUN
C            ELEVATION ANGLE IS 30 DEG. AND INCREASING AND
C            30 DEG. AND DECREASING (OR ANY SIMILAR PAIR),
C            SOLVE FOR THE SUN ELEVATION ANGLE AT THE GIVEN
C            VALUE OF THETA.
C
C
      RTOD=180./3.1415927
      NOSUCH=0
      IF(TNOON.GE.360.)TNOON=TNOON-360.
      C=ACOS(COS(TNOON/RTOD)*COS(BETA/RTOD))
      X=ACOS(TAN(TNOON/RTOD)/TAN(C))
      Z=COS(THETA/RTOD)*COS(C)+SIN(THETA/RTOD)*SIN(C)*COS(X)
      IF(ABS(Z).GT.1.)GO TO 30
      EL=90.-(ACOS(Z)*RTOD)
      RETURN
   30 NOSUCH=1
      EL=0.
      RETURN
      END
```

```
E.THEMAX
                                                  08 MAR 72  01:55:29  CYCLE:  0
C
      SUBROUTINE THEMAX(ARCPAS,MCATEG,THMAX,THMIN,WGTMXC,WGTMXU)
C
      INCLUDE BLK1,LIST
      INCLUDE BLK3,LIST
      INCLUDE BLK5,LIST
      COMMON/PRNT/IPRINT
      DIMENSION TMAX(3,250)
      INTEGER THETA(400), OP1, OP2
C
      NTOTAL = 0
      NMAX = 0
      ISTART = 0
      DO 5 I = 1,3
      DO 5 J = 1,250
    5 TMAX(I,J) = 0
      IF(THMIN .NE. 0 .OR. THMAX .NE. 0) GO TO 10
      BEGIN = THMIDE
      XEND = THMADE
      GO TO 15
   10 BEGIN = THMIN
      XEND = THMAX
C
C   BUILD THETA VECTOR
C
   15 DO 100 J=1,NSCORE
      MISS = SCORE(6,J)
      IF(MCATEG .EQ. 234) GO TO 80
      IF(MCATEG .LE. 4 .AND. MISS .NE. MCATEG) GO TO 100
      OP1 = MCATEG/10
      OP2 = MOD(MCATEG,10)
      IF(MISS .EQ. MCATEG .OR. MISS .EQ. OP1 .OR. MISS .EQ. OP2)
     . GO TO 80
      GO TO 100
   80 CALL PASBAD(SCORF(2,J),SCORE(3,J),BEGIN,XEND,IGOOD)
      IF(IGOOD .EQ. 0) GO TO 100
      NTOTAL = NTOTAL + 1
      THETA(NTOTAL) = J
  100 CONTINUE
C
C   PUT DATA IN ASCENDING ORDER BY THETA
C
  110 IF(NTOTAL .NE. 0) GO TO 112
      WRITE(6,1500)
      RETURN
  112 ISTART = ISTART + 1
      L = THETA(ISTART)
      DO 115 J=ISTART,NTOTAL
      K=THETA(J)
      IF(SCORE(2,L) .LE. SCORE(2,K)) GO TO 115
      THETA(ISTART) = K
      THETA(J) = L
      L = K
  115 CONTINUE
      IF(ISTART .NE. NTOTAL) GO TO 112
C
```

```
C     EVALUATE THE INTERVALS
 120  NMAX = NMAX + 1
      K = THETA(NMAX)
      RBEGIN = SCORE(2,K)
      REND = RBEGIN + ARCPAS
      IF(REND .GT. XEND) GO TO 220
      TMAX(3,NMAX) = RBEGIN
      DO 200 K=NMAX,NTOTAL
      N = THETA(K)
      CALL PASBAD(SCORE(2,N),SCORE(3,N),RBEGIN,REND,IGOOD)
      IF(IGOOD .EQ. 0) GO TO 200
      TMAX(1,NMAX) = TMAX(1,NMAX) + SCORE(5,N)
      TMAX(2,NMAX) = TMAX(2,NMAX) + SCORE(4,N)
 200  CONTINUE
      IF(NMAX .NE. NTOTAL) GO TO 120
 220  NMAX = NMAX - 1
      WGTMXC = TMAX(1,1)
      WGTMXU = TMAX(2,1)
      THSTRT = TMAX(3,1)
      DO 300 K=1,NMAX
      IF(AMAX1(WGTMXC,TMAX(1,K)) .EQ. WGTMXC) GO TO 300
      WGTMXC = TMAX(1,K)
      WGTMXU = TMAX(2,K)
      THSTRT = TMAX(3,K)
 300 CONTINUE
      THEND = THSTRT + ARCPAS
      DO 400 K=1,NMAX
 400  TMAX(1,K) = 0
      WC = WGTMXC
      WU = WGTMXU
      IF(IPRINT .EQ. 1) RETURN
      WRITE(6,2000) IREV, THSTRT, THEND, WC, WU
      RETURN
1500  FORMAT(//T2,'THERE ARE NO ENTRIES IN SCORE IN THE ',
    . 'SPECIFIED MISSION CATEGORY.')
2000  FORMAT(/T2,'REV ',I4,3X,'THSTRT = ',F8.3,3X,'THEND = ',F8.3/
    . T13,'WGTMXC = ' ,F8.3,2X, 'WGTMXU = ',F8.3)
      END
```

```
E.TIMDAT                                                          26 JAN 72  14:12:52  CYCLE:  0
          SUBROUTINE TIMDAT(TIME,MO,IDAY,HR)
C
C    PURPOSE:   CONVERTS TIME EXPRESSED AS TIME ELAPSED FROM
C               MIDNIGHT ON DEC 31 TO MONTH, DAY, AND HOUR
C
          IF(TIME .GE. 120) GO TO 40
          IF(TIME .GE. 31) GO IO 10
          MO = 1
          DUM = TIME
          GO TO 150
     10   IF(TIME .GE. 59) GO TO 20
          MO = 2
          DUM = TIME - 31
          GO TO 150
     20   IF(TIME .GE. 90) GO TO 30
          MO = 3
          DUM = TIME - 59
          GO TO 150
     30   MO = 4
          DUM = TIME - 90
          GO TO 150
     40   IF(TIME .GE. 151) GO TO 50
          MO = 5
          DUM = TIME - 120
          GO TO 150
     50   IF(TIME .GE. 181) GO TO 60
          MO = 6
          DUM = TIME - 151
          GO TO 150
     60   IF(TIME .GE. 212) GO TO 70
          MO = 7
          DUM = TIME - 181
          GO TO 150
     70   IF(TIME .GE. 243) GO TO 80
          MO = 8
          DUM = TIME - 212
          GO TO 150
     80   IF(TIME .GE. 273) GO TO 90
          MO = 9
          DUM = TIME - 243
          GO TO 150
     90   IF(TIME .GE. 304) GO TO 100
          MO = 10
          DUM = TIME - 273
          GO TO 150
    100   IF(TIME .GE. 334) GO TO 110
          MO = 11
          DUM = TIME - 304
          GO TO 150
    110   MO = 12
          DUM = TIME - 334
    150   IDAY = DUM
          HR = (DUM - IDAY) * 24
          IDAY = IDAY + 1
          RETURN
          END
```

```
E.TORGI
        DIMENSION EREP(15,1000)
        NAMELIST/INEREP/EREP
        READ(5,INEREP)
        DO 100 K = 1,1000
        IF(EREP(1,K) .LT. -9000) GO TO 200
        IF(EREP(1,K) .LT. 0) GO TO 100
        NUMEX = EREP(1,K)
        WGTEX = EREP(15,K)
        ISTART = K + 1
        DO 90 J = ISTART, 1000
        IF(EREP(1,J) .LT. -9000) GO TO 95
        IF(EREP(1,J) .LT. 0) GO TO 90
        NUM = EREP(1,J)
        IF(NUM .NE. NUMEX) GO TO 90
        EREP(1,J) = -EREP(1,J)
        WGTEX = WGTEX + EREP(15,J)
   90   CONTINUE
   95   WRITE(6,1000) NUMEX, WGTEX
  100   CONTINUE
 1000   FORMAT(/T2,'EXPERIMENT NUMBER:',I5,5X,
      . 'TOTAL WEIGHT = ',F8.4)
  200   END
```

```
E.VROTAT                                          26 JAN 72  14:13:43  CYCLE:  0
C
      SUBROUTINE VROTAT (V, R, THETA, VNEW)
C
C  THE PURPOSE OF THIS SUBROUTINE IS TO ROTATE AN ARBITRARY VECTOR V(3)
C  ABOUT AN ARBITRARY AXIS R BY AN ANGLE OF THETA, GIVEN IN DEGREES.
C  THE ROTATION AXIS IS DESCRIBED BY THE UNIT VECTOR R(3).  THE
C  RESULTING VECTOR IS GIVEN BY VNEW(3).
C
      DIMENSION V(3), R(3), VNEW(3), B(3), C(3), D(3), DSINT(3),
     •          CCOST(3), DANDC(3)
C
      SINTHE = SIN (THETA * .17453293E-01)
      COSTHE = COS (THETA * .17453293E-01)
C
C  B IS THE COMPONENT OF V ALONG R
C
       VDOTR=VDOT(V,R)
       B(1)=VSCALR(R,VDOTR)
C
C  C IS THE COMPONENT OF V PERPENDICULAR TO R
C
       C(1)=VSUB(V,B)
C
C  D COMPLETES A RIGHT HANDED ORTHOGONAL COORDINATE SET
C
       D(1)=VCROSS(R,C)
C
C  VNEW, THE RESULTING ROTATED VECTOR, IS GIVEN BY D * SIN(THETA) + C *
C   COS(THETA) + B
C
       DSINT(1)=VSCALR(D,SINTHE)
       CCOST(1)=VSCALR(C,COSTHE)
       DANDC(1)=VADD(DSINT,CCOST)
       VNEW(1)=VADD(DANDC,B)
C
      RETURN
      END
```

```
E.WGTADD
          SUBROUTINE WGTADD

          PURPOSE:  SIMULATE THE EFFECT OF HAVING PERFORMED AN EREP PASS
                  BY ADDING TO TOTSCR THE TOTAL WEIGHT ACCOMPLISHED,
                  RECORDING THE DATE THE PASS WAS MADE IF REPEATING
                  PASSES ARE REQUIRED, AND RECORDING VALUE OF OMEGA IF
                  COMPLETE COVERAGE IS REQUIRED

          INCLUDE BLK2,LIST
          INCLUDE BLK3,LIST
          INCLUDE BLK5,LIST

          IADD = 0
          DO 100 I = 1,NSCORE
          CALL PASBAD(SCORE(2,I),SCORE(3,I),THSTRT,THEND,IGOOD)
          IF(IGOOD .EQ. 0) GO TO 100
          IADD = 1
          NPT = SCORE(1,I)
          TOTSCR(NPT) = TOTSCR(NPT) + SCORE(5,I)
          TOTSCR(NPT) = AMIN1(TOTSCR(NPT),EREP(15,NPT))
          ICOV = EREP(7,NPT)
          ITYPE = MOD(ICOV,10)
          IF(ITYPE .EQ. 0) GO TO 100
          IF(ITYPE .EQ. 1) GO TO 20

       THERE IS CURRENTLY NO LOGIC TO HANDLE TOTAL COVERAGE REQUIREMENTS.

          GO TO 100

       REPEATING COVERAGE

     20   DO 50 K = 1,200
          IF(NPT .EQ. PASDAT(1,K)) GO TO 60
     50   IF(PASDAT(1,K) .EQ. 0) GO TO 55
          WRITE(6,6000)
          RETURN
     55   PASDAT(1,K) = NPT
     60   PASDAT(2,K) = PASDAT(2,K) + 1
          NENTR = PASDAT(2,K)
          NPLACE = NENTR + 2
          PASDAT(NPLACE,K) = TRAJ(4,LINE)
    100   CONTINUE
          IF(IADD .NE. 0) GO TO 120
          S = THSTRT
          E = THEND
          WRITE(6,5000) IREV, S, E
          RETURN
    120   IPASS = IPASS + 1
          PASS(1,IPASS) = IREV
          PASS(2,IPASS) = TRAJ(2,LINE)
          IF(PASS(2,IPASS) .LT. 0) PASS(2,IPASS) = 360. + PASS(2,IPASS)
          PASS(3,IPASS) = TRAJ(4,LINE)
          PASS(4,IPASS) = THSTRT
          PASS(5,IPASS) = THEND
          CALL SUNEL(TRAJ(3,LINE),TRAJ(5,LINE),THSTRT,EL,NOSUCH)
          PASS(6,IPASS) = EL
```

```
   57.          CALL SUNEL(TRAJ(3,LINE),TRAJ(5,LINE),THEND,EL,NOSUCH)
   58.          PASS(7,IPASS) = EL
   59.          RETURN
   60.C
   61.C
   62. 5000   FORMAT(/T2,'SUBROUTINE WGTADD--FOR IREV = ',I4,' AND'/
   63.      . T2,'THSTRT = ',F8.3,',',' THEND = ',F8.3,'  NO ENTRIES IN ',
   64.      . 'SCORE ARE ACCEPTABLE.')
   65. 6000   FORMAT(/T2,'ENTRIES IN PASDAT EXCEED DIMENSION*****ERROR')
   66.          END

END CULL  -  DATE: 04/14/72  TIME: 15:33:19  CARD COUNT: 2166  LABEL COUNT: 0  ITEM COUNT: 0




@PRT,T E.
FURPUR 0023-04/14-15:33

EREP*EREP   ELEMENT TABLE

D  NAME         VERSION         TYPE        DATE        TIME      SEQ #  SIZE-PRE,TEXT (CYCLE WORD) PSRMODE  LOCATION
   ELTHET                     FOR SYMB    26 JAN 72   14:00:19      1            6     5    0    1             1792
   ELTHET                     RELOCATABLE 26 JAN 72   14:00:26      2       1    5                             1798
   SUNEL                      FOR SYMB    26 JAN 72   14:11:36      3            6     5    0    1             1804
   SUNEL                      RELOCATABLE 26 JAN 72   14:11:40      4       1    7                             1810
   TIMDAT                     FOR SYMB    26 JAN 72   14:12:52      5           11     5    0    1             1818
   TIMDAT                     RELOCATABLE 26 JAN 72   14:12:58      6       1   11                             1829
   VROTAT                     FOR SYMB    26 JAN 72   14:13:43      7            8     5    0    1             1841
   VROTAT                     RELOCATABLE 26 JAN 72   14:13:50      8       2    5                             1849
   SLOPE                      FOR SYMB    28 JAN 72   14:15:21      9            2     5    0    1             1856
   SLOPE                      RELOCATABLE 28 JAN 72   14:15:27     10       1    2                             1858
   LLIST                      SYMBOLIC    28 JAN 72   15:02:42     11            2     5    0    1             1861
   LONRAN                     SYMBOLIC    01 FEB 72   12:29:09     12           10     5    0    1             1863
   LATLON                     SYMBOLIC    01 FEB 72   11:00:39     13            4     5    0    1             1873
   LONRAN                     RELOCATABLE 01 FEB 72   13:45:48     14       1   14                             1877
   LATLON                     RELOCATABLE 01 FEB 72   13:46:12     15       1    6                             1892
   STCOMP                     SYMBOLIC    04 FEB 72   08:23:56     16           11     5    0    1             1899
   RECOMP                     SYMBOLIC    11 FEB 72   08:36:25     17            6     5    0    1             1910
   FILES                      SYMBOLIC    22 FEB 72   12:32:37     18            3     5    0    1             1916
   CINOTE                     ELT SYMB    29 FEB 72   10:52:23     19           21     5    0    1             1919
   COM                        FOR PROC    07 MAR 72   22:32:40     20            6     1    0    1             1940
   ALLDON                     SYMBOLIC    08 MAR 72   01:44:32     21           13     0    0    1             1946
   ALLDON                     RELOCATABLE 08 MAR 72   01:44:34     22       2   12                             1959
   COMMON                     SYMBOLIC    08 MAR 72   01:45:40     23           24     0    0    1             1973
   COMMON                     RELOCATABLE 08 MAR 72   01:45:43     24       2   26                             1997
   MAIN                       SYMBOLIC    08 MAR 72   01:48:39     25           46     0    0    1             2025
   MAIN                       RELOCATABLE 08 MAR 72   01:48:42     26       4   49                             2071
   NONDON                     SYMBOLIC    08 MAR 72   01:51:24     27           11     0    0    1             2124
   NONDON                     RELOCATABLE 08 MAR 72   01:51:26     28       2   10                             2135
   PASBAD                     SYMBOLIC    08 MAR 72   01:51:27     29            7     0    0    1             2147
   PASBAD                     RELOCATABLE 08 MAR 72   01:51:29     30       1    6                             2154
   PPASS                      SYMBOLIC    08 MAR 72   01:51:31     31            7     0    0    1             2161
   PPASS                      RELOCATABLE 08 MAR 72   01:51:33     32       2    8                             2168
   PPROP                      SYMBOLIC    08 MAR 72   01:51:35     33           13     0    0    1             2178
   PPROP                      RELOCATABLE 08 MAR 72   01:51:37     34       2   11                             2191
   PSCRMC                     SYMBOLIC    08 MAR 72   01:54:02     35           15     0    0    1             2204
   PSCRMC                     RELOCATABLE 08 MAR 72   01:54:04     36       2   17                             2219
   PTSCRM                      SYMBOLIC     08 MAR 72    01:54:07      37              13    0    0    1          2238
   PTSCRM                      RELOCATABLE  08 MAR 72    01:54:09      38       2      14                          2251
   PTSCRP                      SYMBOLIC     08 MAR 72    01:54:12      39              11    0    0    1          2267
   PTSCRP                      RELOCATABLE  08 MAR 72    01:54:13      40       2      11                          2278
   THEMAX                      SYMBOLIC     08 MAR 72    01:55:29      41              20    0    0    1          2291
   THEMAX                      RELOCATABLE  08 MAR 72    01:55:31      42       2      19                          2311
   WGTADD                      SYMBOLIC     08 MAR 72    09:12:19      43              16    5    0    1          2332
   WGTADD                      RELOCATABLE  08 MAR 72    09:12:40      44       2      14                          2348
   FOSCAN                      SYMBOLIC     08 MAR 72    09:42:32      45              58    5    0    1          2364
   FOSCAN                      RELOCATABLE  08 MAR 72    09:42:50      46       3      52                          2422
   MASCAN                      SYMBOLIC     08 MAR 72    15:17:38      47              25    5    0    1          2477
   MASCAN                      RELOCATABLE  08 MAR 72    15:17:49      48       2      27                          2502
   MISION                      SYMBOLIC     14 MAR 72    13:07:17      49              14    5    0    1          2531
   MISION                      RELOCATABLE  14 MAR 72    13:07:19      50       2      13                          2545
   FLYOVR                      SYMBOLIC     16 MAR 72    14:58:18      51              50    5    0    1          2560
   FLYOVR                      RELOCATABLE  16 MAR 72    14:56:43      52       2      46                          2610
   TOWGT                       SYMBOLIC     20 MAR 72    09:24:33      53               5    5    0    1          2658
   TOWGT                       RELOCATABLE  20 MAR 72    09:24:50      54       1       6                          2663
   MSORT                       SYMBOLIC     20 MAR 72    09:46:40      55              30    5    0    1          2670
   MSORT                       RELOCATABLE  20 MAR 72    09:46:51      56       2      34                          2700
   CINOTE                      RELOCATABLE  20 MAR 72    13:26:31      57       1       1                          2736
   ERMAP                       SYMBOLIC     20 MAR 72    13:27:11      58               1    5    0    1          2738
*  ERMAP                       ABSOLUTE     20 MAR 72    13:28:07      59             488                          2739
   SITES                       SYMBOLIC     22 MAR 72    15:55:21      60              25    5    0    1          3227
   SITES                       RELOCATABLE  22 MAR 72    15:55:44      61       2      25                          3252
   ERMAP                       ABSOLUTE     22 MAR 72    15:56:52      62             488                          3279
NEXT AVAILABLE LOCATION-                                                                                           3767

ASSEMBLER PROCEDURE TABLE EMPTY

COBOL PROCEDURE TABLE EMPTY

                    FORTRAN PROCEDURE TABLE

D NAME          LOCATION          LINK    D NAME          LOCATION          LINK    D NAME          LOCATION          LINK
  BLK1                54322        20       BLK2                54351        20       BLK3                54388        20
  BLK4                54421        20       BLK5                54451        20

                    ENTRY POINT TABLE

  D NAME          LINK    D NAME          LINK    D NAME          LINK    D NAME          LINK    D NAME          LINK
    ALLDON          22      CINOTE          57      COMMON          24      ELTHET           2      FLYOVR          52
    FORMAIN$        61      FORMAIN$        56      FORMAIN$        54      FORMAIN$        50      FORMAIN$        26
    FOSCAN          46      LATLON          15      LONRAN          14      MASCAN          48      NONDON          28
    PASBAD          30      PPASS           32      PPROP           34      PSCRMC          36      PTSCRM          38
    PTSCRP          40      SLOPE           10      SUNEL            4      THEMAX          42      TIMDAT           6
    VROTAT           8      WGTADD          44

 FIN
```

```
COCHRAN - CULL COREY FILE EREP.EREP                                                     DATE 041472        PAGE    58

    TIME: 00:00:14.359    IN: 7      OUT: 0       PAGES:   56

    INITIATION  TIME:   15:30:36-APR 14,1972

    TERMINATION TIME:   15:33:22-APR 14,1972

    COR-BLK-SECS: 1335

    IO COUNT:      77

    CHARGE:     1.519
```

APPENDIX B

PROGRAM OUTPUT EXAMPLES

EXAMPLES OF "SIGN-ON" PROCEDURE, AND OF "evaluate" AND "maxwgt" OPTIONS

```
@asg,ax  erep
READY
@add  erep.files
READY
READY
READY
READY
READY
READY
READY
READY
READY
READY
READY
READY
@asg,ax ersav
READY
@asg,a  ersav2
READY
@use  20,ersav
READY
@use  21,ersav2
READY
@xqt  erep.ermap

 CURRENT VALUES OF CONTROL VARIABLES ARE:
 APASDE =   60.0000  MCATDE = 234        MISION =   2
 THMIDE =  -90.0000  THMADE =  270.0000  PERWT =   1.000
 IERSAV =   0        IEROLD =   0        IDEVIC = TERMNL
 ARCMIN =  1.000     OK?
yes


 FIRST REV =   16.     LAST REV =  440.
 FIRST REV WITH ABS BETA GREATER THAN 50:     0
 LAST REV WITH ABS BETA GREATER THAN 50:     0

 OPTION =
evaluate

 REV NO.?     (I4 FORMAT)
0061

 OPTION =
maxwgt

 ARCPAS =   60.0000     MCATEG = 234
 THMIN =   -90.0000     THMAX =  270.0000     O.K.?
yes

 REV    61   THSTRT =   22.483   THEND =   82.483
             WGTMXC =  697.163   WGTMXU =  697.163

 OPTION =
```

EXAMPLE OF PRINT OPTION "prpsum"

```
print
 PRINT OPTION =
prpsum

 TYPE PROPOSAL NUMBER     (I4 FORMAT)
0416


 PROPOSAL SUMMARY, REV =    0
 PROP    FO    PASS COND  PASS UNCOND   ACCUM   TOT POS
  416     1        .0           .0       11.1     33.3
  416     2        .0           .0       33.3     33.3
  416     3        .0           .0       20.4     33.3
  416  TOTAL       .0           .0       64.8    100.0

 OPTION =
```

EXAMPLE OF PRINT OPTION "prgsum"

```
prgsum


% WEIGHT ACCOMPLISHED SO FAR:
CATEG.   ACCUM WGT   TOT WT POS     NON ZERO POS     TOT POS
  2        767.1       1883.2             49           146
  23       383.3        697.4              7            18
  24          .0         50.0              0             2
 234      3437.4       8352.3            124           335
   3       480.1       1698.2             27           152
  34       143.3        683.3              4            19
   4          .0       2002.6              0           185
TOTAL     5211.3      15367.1            211           857

OPTION =
```

EXAMPLE OF PRINT OPTION "pscore"

```
print
 PRINT OPTION =
pscore

 EREP PTR     THETA S     THETA E     WGT UNC     WGT CON      CATEG
     1.00       48.92       55.86       14.29       14.29       2.00
    70.00       48.48       53.71        6.67        6.67       2.00
    83.00       52.57       52.86       10.00       10.00       2.00
    84.00       22.64       29.99       50.00       50.00       2.00
    97.00       51.22       52.64       33.33       33.33       2.00
    98.00       60.66       79.96        6.67        6.67       2.00
   100.00       26.08       39.80        6.67        6.67       2.00
   118.00       51.53       52.25       33.33       33.33       2.00
   142.00       22.48       40.75        4.35        4.35       2.00
   163.00       46.40       52.36       50.00       50.00      23.00
   167.00       48.92       55.86       50.00       50.00     234.00
   201.00       51.15       52.21       11.11       11.11     234.00
   203.00       45.37       61.37        1.85        1.85     234.00
   252.00       48.92       55.86       50.00       50.00     234.00
   262.00       48.92       55.86       25.00       25.00     234.00
   266.00       48.92       55.86       25.00       25.00     234.00
   271.00       50.46       52.67       25.00       25.00     234.00
   295.00       47.15       47.18       10.00       10.00     234.00
   315.00       51.49       52.15       10.00       10.00     234.00
   345.00       51.23       52.90        5.88        5.88     234.00
   364.00       33.48       34.92        8.33        8.33     234.00
   365.00       33.48       34.92        8.33        8.33     234.00
   368.00       33.48       39.26        8.33        8.33     234.00
   369.00       33.48       39.26        8.33        8.33     234.00
   372.00       50.50       57.34        8.33        8.33     234.00
   376.00       26.52       40.75       25.00       25.00     234.00
   377.00       51.05       89.02       25.00       25.00     234.00
   388.00       -8.65         .00       50.00       50.00     234.00
   394.00       37.80       50.73       16.67       16.67     234.00
   402.00       51.24       52.90        5.00        5.00     234.00
   428.00       51.49       52.27        7.69        7.69     234.00
   432.00       51.24       52.90       50.00       50.00     234.00
   433.00       50.46       52.67       47.00       47.00     234.00
   480.00       37.80       50.73       50.00       50.00     234.00

 OPTION =
```

EXAMPLE OF PRINT OPTION "revsum"

```
print
 PRINT OPTION =
revsum

 THSTRT =    22.4826      THEND =    82.4826      O.K.?
yes

 SUMMARY OF REV  61, THSTRT =  22.5, THEND =  82.5

 MCATEG  WGT COND  WGT UNC  ACCUM  TOT WGT POS   NO FOS
    2     165.3     165.3     .0     165.3          9
   23      50.0      50.0     .0      50.0          1
   24        .0        .0     .0        .0          0
  234     481.9     481.9     .0     753.9         24
 TOTAL    697.2     697.2     .0     969.2         34

 OPTION =
```

EXAMPLE OF PRINT OPTION "revsum"

```
p
 PRINT OPTION =
revsum

 THSTRT =  165.6282     THEND =  225.6282     O.K.?


 SUMMARY OF REV ***, THSTRT = 165.6, THEND = 225.6

 NCATEG  WGT COND  WGT UNC  ACCUM  TOT WGT POS   NO POS
   23        .0       .0      .0        .0          0
  234     262.5    287.5   100.0     412.5         11
    3      22.4     73.3    54.3      76.7         10
   34        .0       .0      .0        .0          0
 TOTAL    284.9    360.8   154.3     489.2         21

 OPTION =
```

EXAMPLE OF PRINT OPTION "missum"

```
print
 PRINT OPTION =
missum

 SUMMARY OF ACCUMULATED SCORE FOR SL3 MISSION
 MISSION CAT    WGT SAT  TOT WGT POS  NON ZERO FOS  TOT FOS
     23          383.3      697.4           7          18
    234         3437.4     8352.3         124         335
      3          480.1     1698.2          27         152
     34          143.3      683.3           4          19
    TOTAL       4444.1    11431.3         162         524

 OPTION =
```

EXAMPLE OF PRINT OPTION "eresum"

```
p
 PRINT OPTION =
eresum


 PROPOSAL SUMMARY, REV = 2104
 PROP      FO     PASS COND   PASS UNCOND      ACCUM      TOT POS
  352    TOTAL        .0            .0          10.6       100.0
  399    TOTAL        .0            .0         100.0       100.0
 5472    TOTAL        .0            .0         100.0       100.0
  455    TOTAL        .0            .0         100.0       100.0
  363    TOTAL        .0            .0          40.0       100.0
  446    TOTAL        .0            .0          12.5       100.0
  523    TOTAL        .0            .0            .0       100.0
  582    TOTAL        .0            .0          33.3       100.0
  373    TOTAL        .0            .0            .0       100.0
  485    TOTAL        .0            .0         100.0       100.0
  402    TOTAL        .0            .0            .0       100.0
 2540    TOTAL        .0            .0         100.0       100.0
 8540    TOTAL        .0            .0          14.3       100.0
 3540    TOTAL        .0            .0            .0       100.0
 6540    TOTAL        .0            .0            .0       100.0
  416    TOTAL        .0            .0          54.8       100.0
  386    TOTAL        .0            .0          25.0       100.0
   69    TOTAL        .0            .0          55.6       100.0
 5540    TOTAL        .0            .0            .0       100.0
  345    TOTAL        .0            .0         100.0       100.0
   23    TOTAL        .0            .0            .0       100.0
  390    TOTAL        .0            .0            .0       100.0
  352    TOTAL        .0            .0          10.6       100.0
  531    TOTAL      50.0          50.0            .0       100.0
  117    TOTAL        .0            .0         100.0       100.0
 3407    TOTAL        .0            .0            .0       100.0
  418    TOTAL        .0            .0           5.6       100.0
  473    TOTAL        .0            .0          50.0       100.0
  528    TOTAL     100.0         100.0            .0       100.0
 3508    TOTAL        .0            .0            .0       100.0
 4508    TOTAL        .0            .0            .0       100.0
  212    TOTAL        .0            .0         100.0       100.0
  323    TOTAL        .0            .0            .0       100.0
  358    TOTAL        .0            .0          50.0       100.0
  385    TOTAL        .0            .0            .0       100.0
  443    TOTAL        .0            .0            .0       100.0
  459    TOTAL        .0            .0          50.0       100.0
  496    TOTAL      12.5          12.5            .0       100.0
  497    TOTAL        .0            .0          75.0       100.0
  498    TOTAL        .0            .0          75.0       100.0
  591    TOTAL        .0            .0          75.0       100.0
  558    TOTAL        .0            .0         100.0       100.0
  325    TOTAL        .0            .0            .0       100.0
  332    TOTAL        .0            .0            .0       100.0
  349    TOTAL        .0            .0            .0       100.0
  380    TOTAL        .0            .0            .0       100.0
  409    TOTAL        .0            .0         100.0       100.0
  423    TOTAL        .0            .0         100.0       100.0
  434    TOTAL        .0            .0          50.0        50.0
  437    TOTAL        .0            .0          50.0       100.0
  491    TOTAL        .0            .0         100.0       100.0
  495    TOTAL        .0            .0         100.0       100.0
  537    TOTAL        .0            .0          60.0       100.0
  527    TOTAL      50.0          50.0            .0       100:0
  530    TOTAL        .0            .0            .0       100.0
  541    TOTAL        .0            .0          70.0       100.0
  557    TOTAL        .0            .0            .0       100.0
```

EXAMPLE OF PRINT OPTION "aldone"

```
print
 PRINT OPTION =
aldone


 COMPLETED F.O.S

 PROP   FO    MCAT   POS SCR    ACCUM SCR
 1540    1     2      14.29      14.29
  486    8     2       7.69       7.69
  486   12     2       7.69       7.69
  562    1     2      50.00      50.00
  410    1     2      33.33      33.33
  352   20     2       2.44       2.44
  382    1     2      16.67      16.67
  452    1     2       5.55       5.55
  452    3     2       5.55       5.55
  452    5     2       5.55       5.55
  473    1     2      50.00      50.00
  529    5     2       2.38       2.38
  499    6     2       6.67       6.67
  499    7     2       6.67       6.67
  499    8     2       6.67       6.67
  499    9     2       6.67       6.67
  499   10     2       6.67       6.67
  520    1     2      11.10      11.10
 3009    1     2      50.00      50.00
  393    1     2      33.33      33.33
  491    7     2      10.00      10.00
  438    1     2      50.00      50.00
  463    1     2      33.33      33.33
  474    1     2      50.00      50.00
  504    1     2      25.00      25.00
  504    3     2      25.00      25.00
  471    1     2      33.33      33.33
  517    1     2      33.33      33.33
  550    1     2       6.67       6.67
  550    2     2       6.67       6.67
  550    3     2       6.67       6.67
  550    5     2       6.67       6.67
 3313    1     2      33.33      33.33
  398    1     2      25.00      25.00
  417    1     2      33.33      33.33
  454   12     2       2.17       2.17
  454   14     2       2.17       2.17
  454   18     2       2.17       2.17
  454   36     2       1.45       1.45
  535    4     2       6.25       6.25
  599    1     2       4.35       4.35
  599   13     2       4.35       4.35
  399    1    23     100.00     100.00
 5472    1    23     100.00     100.00
  455    1    23     100.00     100.00
  363    2    23      20.00      20.00
  440    5    23      12.50      12.50
  485    1    23      50.00      50.00
 2549    1   234     100.00     100.00
```

EXAMPLE OF PRINT OPTION "nowork"

```
print
 PRINT OPTION =
nowork


 F.O.S ON WHICH NO WORK HAS BEEN DONE
 PROP  FO    MCAT    POS SCR
  486   1     2       7.69
  486   4     2       7.69
  486   6     2       7.69
  486  10     2       7.69
 1457   1     2      33.33
 1457   2     2      33.33
 2457   1     2      50.00
 4457   1     2     100.00
  475  13     2       4.76
  475  16     2       4.76
  475  19     2       4.76
  475  10     2       4.76
  475   1     2       4.76
  475   4     2       4.76
  475   7     2       4.76
   52  21     2       3.33
```

COMPLETE PRINTOUT WAS INTERRUPTED AT THIS POINT.

```
 SUMMARY
 MISSION CAT    TOT POS SCR    NO OF FOS
     2            1075.8          97
    23             312.5          11
    24              50.0           2
   234            4493.2         211
  TOTAL           5931.5         321

 OPTION =
 OPTION           IS NOT VALID.

 OPTION =
```

EXAMPLE OF PRINT OPTION "passum"

```
print
 PRINT OPTION =
passum


   31 PASSES HAVE BEEN ACCEPTED

                                PASS STARTS                                     SUN ELEVATION
 REV NO       OMEGA      MO  DAY    HR          THSTRT       THEND      START          END
    61       248.03       5    4   16.2122     22.48        82.48     58.45         48.34
    77       228.79       5    5   17.0753     30.84        90.84     58.14         44.93
    93       209.56       5    6   17.9382     40.13       100.13     57.98         41.26
   108       214.04       5    7   17.2472     37.80        97.80     51.72         45.58
   123       218.51       5    8   16.5560     37.80        97.80     46.64         49.81
   137       246.69       5    9   14.3108     65.36       125.36     56.86         33.98
   153       227.46       5   10   15.1733     31.80        91.80     31.80         56.96
   169       208.24       5   11   16.0357     41.40       101.40     33.73         56.31
   184       212.72       5   12   15.3441     42.98       102.98     30.00         57.97
   199       217.20       5   13   14.6524     45.31       105.31     27.00         59.45
   213       120.89       5   14   20.1761    106.93       166.93     61.17         29.80
   249       112.16       5   16   20.3460    103.76       163.76     59.62         44.01
   264       110.65       5   17   19.6539     99.38       159.38     54.70         53.18
   294       125.63       5   19   18.2693    107.86       167.86     56.87         57.07
   325       110.91       5   21   18.4383     96.96       156.96     39.84         77.76
   340       115.41       5   22   17.7457    199.32       259.32     41.38         *****
  1778       238.77       8   23   19.8505     33.84        93.84     67.42          7.76
  1793       243.36       8   24   19.1517     29.40        89.40     76.12         16.13
  1854       238.02       8   28   17.9094     34.93        94.93     71.13         23.45
  1869       242.61       8   29   17.2103     -2.82        57.18     47.67         58.49
  1930       237.29       9    2   15.9667     24.69        84.69     43.51         45.20
  1945       241.89       9    3   15.2672     31.58        91.58     41.87         43.10
  2010       141.82       9    7   20.2355     82.28       142.28     44.38         29.75
  2041       127.33       9    9   20.3892    107.81       167.81     47.29         24.69
  2056       131.94       9   10   19.6892    109.69       169.69     48.05         29.16
  2086       141.15       9   12   18.2891    111.40       171.40     48.27         39.56
  2102       122.06       9   13   19.1422    111.27       171.27     47.11         45.79
  2117       126.67       9   14   18.4418    140.35       200.35     59.58         30.31
  2132       131.28       9   15   17.7414    104.34       164.34     37.48         60.81
  2178       121.43       9   18   17.1928    111.78       171.78     35.97         73.65
  2193       126.05       9   19   16.4919    108.81       168.81     29.23         79.45

 OPTION =
```

Subject: A Computer Program to Aid Skylab Earth Resources Experiment Planning

From: D. A. Corey

Distribution List

NASA Headquarters

R. O. Aller/MLO
H. Cohen/MLQ
J. M. DeNoyer/SR
*J. H. Disher/MLD
*J. P. Field, Jr./MLB
T. L. Fischetti/SG
*W. D. Green, Jr./MLA
*T. E. Hanes/MLA
E. L. Keith/MLO
*A. S. Lyman/MAP
*C. W. Mathews/E
A. B. Park/SRR
*M. Savage/MLE
W. C. Schneider/ML

MSC

M. Hagan/FCD
*C. R. Hoss/FM
G. L. Hunt/FM/13
M. P. Kilbourn/FM2
*K. S. Kleinknecht/KA
E. L. Lemons/FM2
E. C. Lineberry/FM6
R. O. Linney/FCD
F. C. Littleton/KM
D. W. Owens/FCD
O. G. Smith/KW
P. J. Stull/TD
H. E. Whitacre/KM
K. L. Young/FM

MSFC

*L. F. Belew/PM-SL-MGR
*C. C. Hagood/S&E-CSE-A
*R. C. Lester/S&E-CSE-MP
*R. E. Tinius/S&E-CSE-MP

Lockheed Electronics Company, Houston Texas

J. G. Baron
R. H. Sutton

Philco, Houston, Texas

J. Nemec

BTL

D. A. De Graaf
D. H. Hagner
*H. Z. Hardaway
*E. E. Sumner
*M. P. Wilson

* Abstract and Introduction Only.